all rights reversed

HACKER

Ιούνης 2012 # Τιμή τεύχους 6,40€

τεύχος 009 - El Pollo Diablo Edition

So you want to be a hacker, eh? [*]



Scurvy Interrupts

LeChuck the Ripper

Bruteforcer in a negligee

pyGame Invaders

Whitehats in Black

[*] Προϋπόθεση να ξεχωρίζετε τους φιγουρατζήδες από τους σοβαρούς, να καταλαβαίνετε το χιούμορ του The Big Bang Theory και να μη μασάτε με όσους επιχειρούν να σας μειώνουν ή/και να σας αποθαρρύνουν. Επίσης, να έχετε συνειδητοποιήσει ότι ο αληθινός hacker δεν σταματά ποτέ να μαθαίνει.

Περιεχόμενα

και οδηγίες χρήσης

58 επιτεθείτε με όοοοολη σας τη δύναμη!

οι δουλειές δεν θέλουν κόπο, θέλουν τρόπο. όταν όμως δεν υπάρχει τρόπος, τότε δεν γλιτώνουμε τον τρόπο. μπερδευτήκατε;

66 πίσω από τις λέξεις, κρύβονται οι συνδέσεις!

DNS εκεί; πες μου για 'σένα, θέλω να μιλήσω μαζί σου...

76 Python Game Programming: Pygame Invaders!

ήρθε η ώρα να φτιάξετε το δικό σας shoot 'em up ;)

88 έξυπνα, γρήγορα κι άνετα κατεβάσματα, από το VPS!

χρησιμοποιείτε ακόμα το VPS σας για web hosting, άντε και για το OpenVPN;! ήρθε η ώρα να δείτε το όλο θέμα πιο... ανοιχτόμυαλα!

Εκδότης / Διευθυντής	Χρήστος Βαρελάς
Αρχισυντάκτης	Παναγιώτης Βαρελάς
Σύμβουλοι έκδοσης	Νίκος Μουρατίδης, Βούλα Παυλίδου
Σύνταξη	Παναγιώτης Βαρελάς, Χρήστος Βαρελάς, Ανδρέας Βενιέρης, Γιώργος Γιάννου, Μανώλης Κιαγιάς, Ιωάννης Κονιάρης, Γιώργος Μελέκος, Νίκος Μουρατίδης, Νικόλαος Σηφάκης, Μέσος Παπαδόπουλος (TM), Χρίστος Τόμπρας
Γραμματική επιμέλεια	Βούλα Παυλίδου
Καλλιτεχνική Επιμέλεια	Πέτρος Φιλιππίδης

Το περιοδικό deltaHacker είναι συνδρομητικό και εκδίδεται από την Parabing Creations δώδεκα φορές το χρόνο, κάθε μήνα. Η Parabing Creations έχει την έδρα της στη Μαιώτιδος 3, 55 133, Καλαμαριά. Η εκτύπωση και η βιβλιοδεσία γίνονται από την ThessPrint A.E. Το email επικοινωνίας του περιοδικού είναι το talk2us@deltahacker.gr και το email για τις συνδρομές το subscriptions@deltahacker.gr. Οι πληροφορίες για τις συνδρομές είναι στο http://deltahacker.gr/subscriptions και οι παραγγελίες γίνονται από το http://deltahacker.gr/order. Όλες οι απόψεις που εκφράζονται στα άρθρα δεν εκφράζουν απαραίτητα και τη γνώμη του περιοδικού.

(*) ψάχνουν ακόμα για μαράκες;

Η καλύτερη δουλειά στον κόσμο!

Πώς δημιουργείς ένα διασκεδαστικό αλλά και πραγματικά χρήσιμο περιοδικό; Ποια θέση παίρνεις απέναντι στα 129 χιλιάδες προβλήματα που σε περιμένουν ήδη με το καλημέρα; Συνέβη κάτι άσχημο σε γνωστή αλυσίδα εδεσματοπωλείων της Θεσσαλονίκης, ή απλά κάνουν ανακαίνιση στο κατάστημα της Καλαμαριάς;

Να σας πω την αλήθεια, τι ακριβώς συμβαίνει --ή συνέβη και τώρα που διαβάζετε αυτές τις γραμμές δεν συμβαίνει-- με το γνωστό εδεσματοπωλείο της Θεσσαλονίκης, ούτε το γνωρίζω ούτε το έχω ψάξει ακόμα. Αυτό που ξέρω είναι ότι τις προάλλες κατέβαινα την Κομνηνών, αυτή στην Καλαμαριά, και καθώς περνούσα μπροστά από το σουβλατζίδικο συνειδητοποίησα ότι ήταν κλειστό, ενώ δεν θα έπρεπε. Στις τζαμαρίες ήταν στερεωμένα κάτι μεγάλα χαρτόνια, μπόρεσα ωστόσο να διακρίνω μια κινητικότητα εντός. Αυτό πάντως που κέρδισε την προσοχή μου --και πραγματικά δεν θα μπορούσε να γίνει διαφορετικά-- ήταν η επιγραφή στη μεγάλη ταμπέλα απ' έξω: "Είτε με ευρώ είτε με δραχμή, εμείς σουβλάκια θα πουλάμε!"

Χαμογέλασα (μάλλον περισσότερο απ' όσο θα έπρεπε) και συνέχισα στο δρόμο μου. Λίγο αργότερα, άγνωστο γιατί, άρχισα να το σκέφτομαι περισσότερο. Θα σας το πω χωρίς περιστροφές -- και γελάστε ελεύθερα: Η ατάκα με τα σουβλάκια και το νόμισμα, μου θύμισε το deltaHacker.

Ένα χρόνο πριν, όταν είχαμε αρχίσει να σκεφτόμαστε και να συζητάμε την πιθανότητα έκδοσης νέου περιοδικού, οι δυσκολίες και τα εμπόδια που βλέπαμε μπροστά μας ήταν αναρίθμητα. Εντάξει, δεν ήταν κι απειραριθμήσιμα, αλλά σίγουρα ήταν πολλά. Κι όχι μόνο αυτό, αλλά όλα τα σημάδια και οι προβλέψεις μάς υποδείκνυαν να καθίσουμε στ' αυγά μας, κατά το κοινώς λεγόμενο, και, τέλος πάντων, ν' αναζητήσουμε κάτι άλλο, ασφαλέστερο. Ένα περίπου χρόνο αργότερα, δύο πράγματα έχουμε συνειδητοποιήσει: **α)** οι δυσκολίες ήταν πολλές περισσότερες και σαφώς μεγαλύτερες απ' ό,τι είχαμε φανταστεί, **β)** κάναμε πολύ καλά που αγνοήσαμε τις δυσκολίες κι εστιάσαμε όλη μας την ενέργεια σ' αυτό που ξέρουμε να κάνουμε καλά και, το κυριότερο, το αγαπάμε πραγματικά.

Όπως σας έλεγα και στο editorial του προηγούμενου τεύχους, όταν θέλεις να πετύχεις σ' ένα δύσκολο περιβάλλον μία από τις επιλογές σου είναι ν' αλλάξεις τους κανόνες του παιχνιδιού. Αν για οποιονδήποτε λόγο δεν μπορείς να τους αλλάξεις, μπορείς πάντα να βγεις από το παιχνίδι και να παίξεις σ' ένα άλλο, στο οποίο όμως εσύ θα 'χεις ορίσει τους βασικούς κανόνες. Τώρα, λογικό είναι να πηγαίνει το μυαλό σας στην πολιτική και συγκεκριμένα στις συζητήσεις για το αν η Ελλάδα πρέπει να μείνει στο ευρώ, στην ευρωζώνη κ.λπ. κ.λπ. Τη στιγμή μάλιστα που θα διαβάζετε τις γραμμές αυτές, είναι πολύ πιθανό πολλά πράγματα να έχουν αλλάξει. Τώρα που το καλοσκέφτομαι, μπορεί και να μην έχουν αλλάξει -- ή τέλος πάντων να έχουν δρομολογηθεί εξελίξεις. Ποιος ξέρει; Εγώ πάντως όχι. Ούτως ή άλλως δεν αναφέρομαι στην πολιτική εδώ, ούτε σ' όλα όσα συμβαίνουν γύρω μας. Σε εντελώς διαφορετικό κλίμα, όλη η κουβέντα που κάνω είναι για έναν χώρο που ξεκινήσαμε να διαμορφώνουμε από το περσινό καλοκαίρι. Πρόκειται για έναν χώρο που ναι μεν σχετίζεται με την πραγματικότητα (είναι γνήσιος υποχώρος της, για να το πω μαθηματικά), εντός των πλαισίων του όμως ένα περιοδικό σαν το deltaHacker πετυχαίνει, βελτιώνεται διαρκώς και συνεχίζει αισιόδοξα!

Πώς δημιουργήσαμε αυτόν το χώρο ευημερίας, για το αγαπημένο μας περιοδικό; Υπήρχαν κάποια μυστικά; Κάποιες βασικές προϋποθέσεις, ίσως; Για μυστικά δεν ξέρω, κάποιες προϋποθέσεις όμως υπήρχαν και πάντα υπάρχουν...

Κέφι, ζωντάνια, παιχνίδι

Αλλάζοντας λίγο το σλόγκαν μιας παλιάς διαφήμισης, στο deltaHacker πρώτα ζούμε και μετά γράφουμε! Πιο συγκεκριμένα, μας αρέσει ο προγραμματισμός, το χάκινγκ, τα δίκτυα και η τεχνολογία γενικότερα. Για κάθε άρθρο που φιλοξενούμε στο περιοδικό ή στο site, έχουν προηγηθεί η εξερεύνηση, οι δοκιμές, το παιχνίδι, οι πειραματισμοί και φυσικά οι απαραίτητες επαληθεύσεις. Το γράψιμο του άρθρου μένει πάντα τελευταίο και μπορώ να σας πω ότι γίνεται με πολύ κέφι, αφού βιαζόμαστε να σας μιλήσουμε για ό,τι μας κράτησε απασχολημένους *πριν* το άρθρο -- και πολύ συχνά μας κρατά και μετά.

Διαρκής διάλογος

Αφού μας αρέσει ό,τι κάνουμε, ε, δεν είναι λογικό κι αναμενόμενο να θέλουμε να μιλάμε και σ' άλλους γι' αυτό; Είναι. Φυσικά, η ευκαιρία για επικοινωνία δεν θέλουμε να μας δίνεται μία φορά κάθε μήνα, μόνο από τις σελίδες του περιοδικού! Αν είναι δυνατόν, δηλαδή. Σε τι εποχή ζούμε; Επειδή λοιπόν ζούμε στην εποχή που ζούμε και μας αρέσει να μιλάμε γι' αυτό που κάνουμε κι αγαπάμε, σας ζαλίζουμε όλη την ώρα από το Twitter (twitter.com/deltaHacker), το Facebook (facebook.com/deltaHacker, το site μας (deltahacker.gr), τα deltaStreams (http://deltahacker.gr/category/canned_streams), τα deltaCasts (coming soon!) και ποιος ξέρει τι άλλο θα σκαρφιστούμε για να τα λέμε :)

Νέο μοντέλο διανομής

Διανομή στο περίπτερο εντύπου με εξειδικευμένο περιεχόμενο, με τις συνθήκες που επικρατούν σήμερα Εκεί Έξω (TM); Α, ευχαριστούμε πάρα πολύ, αλλά δεν θα πάρουμε. Όσοι ενδιαφέρονται πραγματικά για το περιοδικό μπορούν να παραγγέλνουν το τεύχος τους ή τη συνδρομή τους online, από το http://deltahacker.gr/order. Προσφέρουμε αρκετούς τρόπους πληρωμής, ώστε να μπορούν να το παίρνουν ακόμη κι όσοι δεν έχουν PayPal ή/και δεν θέλουν πάρε-δώσε με τράπεζες. Σε κάθε περίπτωση το περιοδικό το ταχυδρομούμε μέσω ΕΛΤΑ. Όλοι μένουν ευχαριστημένοι μ' αυτόν τον τρόπο διάθεσης, ενώ την ίδια στιγμή δεν χάνουμε ούτε ένα ευρώ με το πεπαλαιωμένο μοντέλο διανομής.

Επένδυση στο μέλλον, αγκαλιάζοντας τους πάντες

Το περιοδικό δεν απευθύνεται *μόνο* σε έμπειρους χρήστες ή "φτασμένους" χάκερ. Δεν τους γυρίζει βέβαια την πλάτη, όπως όμως θα 'χετε διαπιστώσει έχει πολλά να προσφέρει σε *όλους* τους φίλους της τεχνολογίας, ανεξαρτήτως της τεχνικής τους κατάρτισης! Γι' αυτό και σε κάθε τεύχος υπάρχει κάτι για όλους. Πιστεύουμε --και η έως τώρα εμπειρία μάς επαληθεύει-- ότι μέσα από χαλαρά, ευχάριστα και ταυτόχρονα πληροφοριακά μεστά και ουσιώδη κείμενα, μπορούμε να σας κάνουμε όλους να ενδιαφέρεστε για πράγματα που ενδιαφέρουν κι εμάς. Την ίδια στιγμή μας δίνετε φρέσκες ιδέες κι όλοι μαζί βελτιωνόμαστε διαρκώς, αναζητώντας τι άλλο έχει νόημα να υλοποιούμε! Αυτό θα πει ουσιαστική επένδυση στο μέλλον, δεν νομίζετε;

Μια τρανταχτή συνειδητοποίηση κι ένα ευχάριστο αποτέλεσμα

Φορές που συζητάμε με συνεργάτες και φίλους, συνειδητοποιούμε ότι έχουμε κάνει το χόμπι μας επάγγελμα. Αισθανόμαστε πολύ τυχεροί γι' αυτό. Η καλύτερη δουλειά του κόσμου είναι αυτή που δεν τη βλέπεις σαν δουλειά. Κι όταν είσαι τυχερός και την έχεις, ε, τότε το αποτέλεσμα είναι τουλάχιστον καλό.

Κι όχι μόνο για 'σένα.

subZraw

POST CAR

THE ADDRESS TO BE WRITTEN ON THIS SIDE





Ουέλ μέιλ

Αγαπητοί μου,

Σας γράφω, επιστρέφοντας από το Προεδρικό Μέγαρο, όπου είχα πάει σήμερα το πρωί προκειμένου να συμπαρασταθώ έμπρακτα στις προσπάθειες δημιουργίας Κυβέρνησης.

Όχι, όχι, μη πάει ο νους σας στο κακό, ο ρόλος μου ήταν αμιγώς θεσμικός.

Θεσμικός υπό την έννοια του ότι ακόμα και αν στέκεσαι απλώς από δίπλα και παρατηρείς, είσαι κι εσύ ένας βασικός θεσμός του κράτους. Ένας πυλώνας, ούτως ειπείν.

Πόσο μάλλον όταν έχουν μόλις γίνει εκλογές, οπότε όλοι στεκόμαστε από δίπλα και παρατηρούμε, γινόμενοι θεσμικοί πυλώνες θέλοντας και μη.

Εγώ ως πυλώνας στεκόμουν απ' έξω, στο ψιλόβροχο, μαζί με τους άλλους εκπροσώπους του Κυρίαρχου Λαού.

Σας γράφω, λοιπόν, έχοντας μόλις επιστρέψει από το Προεδρικό Μέγαρο, και με πλήρη συναίσθηση του κινδύνου να επισύρω την μήνιν του Κυρίου Εκδότου (TM), καθώς προτίθεμαι να πολιτικολογήσω, έστω λιγουλάκι.

Αλλά πείτε μου εσείς. Έχετε ματαδεί τέτοιο μήνυμα εκ μέρους του Κυρίαρχου Λαού; Τέτοιο Ήξεις Αφήξεις; (Άγνωστες λέξεις; Σα δε ντρεπόμαστε. Γκουγκλ ιτ!)

Δηλαδή τι ακριβώς διεμήνυσε ο Κυρίαρχος Λαός, εσείς καταλάβατε; Γιατί εγώ, αν και αποτελώ δομικώς αναπόσπαστο κομμάτι του περί ου ο λόγος Κυρίαρχου Λαού, και στεκόμουν, όπως είπαμε, τόση ώρα στο ψιλόβροχο, μαζί με άλλους εκπροσώπους του, ομολογώ ότι αδυνατώ να το κατανοήσω (το μήνυμα).

Έτσι είναι αν έτσι νομίζετε, που θα έλεγε και ο Πιραντέλο.

Θα έπρεπε, βέβαια να το είχα προβλέψει.

Τα θυμάμαι ακόμα τα μαύρα φίδια που με είχαν ζώσει όταν η Μαιρούλα (της Κυρίας Καίτης, καλέ) μας είχε ανακοινώσει περιχαρής ότι έχει ασπαστεί τον Δωδεκαθεϊσμό.

Αρχικά είχε πάει να με πιάσει απελπισία, μετά το εξέλαβα ως ένδειξη χιούμορ και μου φάνηκε χαριτωμένο μέσα στην μεταμοντερνιά (sic) του, και στο τέλος αποφάσισα να το βγάλω από το μυαλό μου. Δεν έμαθα ποτέ αν το εννοούσε.

Απώθηση το ονομάζει αυτό ο Φρόυντ, το φαινόμενο δηλαδή κατά με το οποίο, κάτι, που σε κανονικές συνθήκες θα μπορούσες να το θυμάσαι και να το ανακαλείς κατά βούληση, παραμένει απροσπέλαστο σε πείσμα όλων των προσπάθειών σου.

Δωδεκαθεϊσμός, σου λέει ο άλλος. Τι διάτανος είναι πάλι τούτος; Το ξεχνάς, λοιπόν, και πας παρακάτω.

Όχι πολύ παρακάτω, ένα δύο στενά μόνο, όσο να γίνουν οι εκλογές -- οπότε βλέπεις αίφνης τα Χρυσά Αβγά (TM) να μαζεύουν κοτζάμ 7%, και ανα-

ρωτιέσαι τι ήταν αυτό κι από πού σου ήρθε.

Με κίνδυνο να εκνευρίσω (και αποξενώσω) το 7% όσων από εσάς με διαβάζουν αυτή τη στιγμή, δεν μπορώ παρά να το σχολιάσω.

Μα καλά, ρε σύντεκνοι, τόση αγανάκτηση πια σας έπιασε που πάθατε και αμνησία; Ναι, σε σας μιλάω, στο 7%. Τα Χρυσά Αβγά βρήκατε να ψηφίσετε; Δηλαδή ελπίζετε σε ένα καλύτερο μέλλον (εικάζω) και πιστεύετε ότι το μέλλον αυτό θα έρθει από κάτι άξεστους φουσκωτούς που δεν ξέρουν να ξεχωρίζουν τον Πλάτωνα από τα πλατάνια, και τις Πλαταιές και απ' τις πλατείες αλλά είναι περήφανοι για όλα τους, αρκεί που είναι Ελληνικά;

Από ανθρώπους που λένε ότι ο Χίτλερ «λατρεύτηκε από το γερμανικό λαό και η χώρα του επί των ημερών του γνώρισε οικονομική άνθηση»;

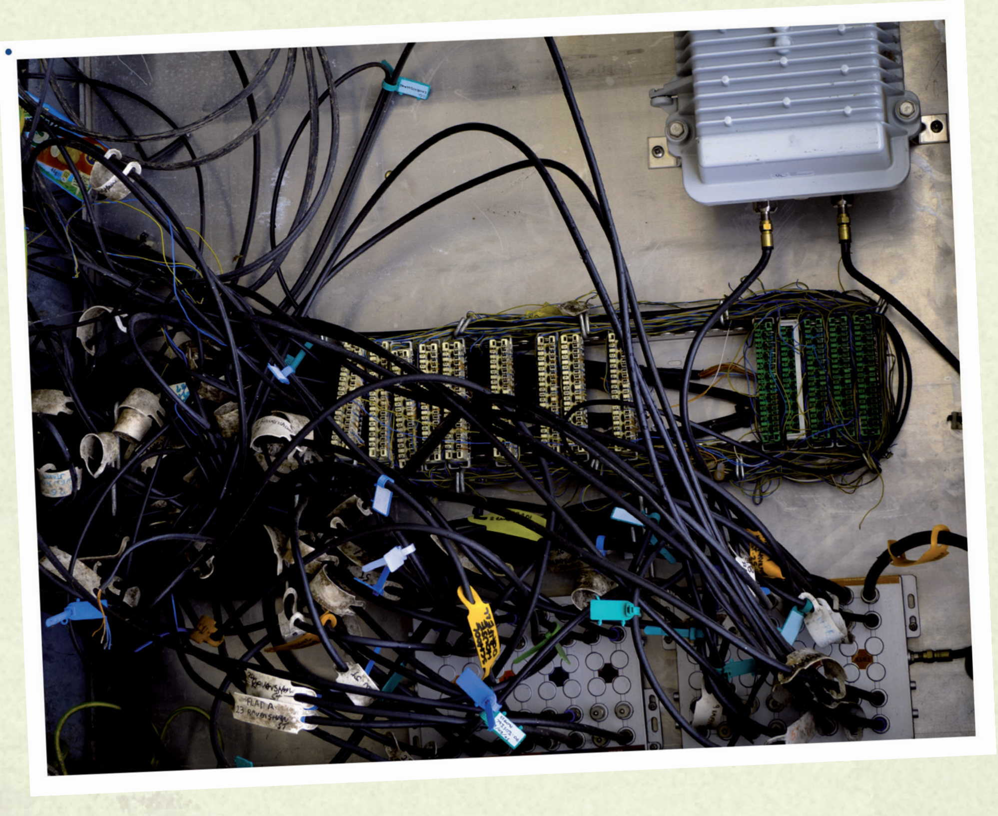
Για όνομα! Το μυαλό μας το 'δωσε, ο Θεούλης, για να το βάζουμε να δουλέψει, όχι για να το περιφέρουμε εδώ κι εκεί ως τεκμήριο ύπαρξης Ανεκμετάλλευτων Πόρων και Χαμένων Ευκαιριών.

Τέλος πάντων. Μόλις επέστρεψα, όπως σας είπα, από το Προεδρικό Μέγαρο όπου οι προσπάθειες δημιουργίας κυβέρνησης εθνικής ενότητας, σωτηρίας ή όπως αλλιώς θέλετε να το πείτε, έχουν αποβεί άκαρπες.

Πράγμα που σημαίνει ότι πάμε ξανά σε εκλογές. (Σ.τ.Ε. Εχουμε πάει ήδη :P)

Είμαι πολύ απογοητευμένος και το μόνο που μου δίνει κάποια αίσθηση προοπτικής και ελπίδας για το μέλλον είναι το Κοτόπουλο Τζαλφρέζι (με ντομάτα και κουρκουμά) που έχουν προγραμματίσει για απόψε η μητέρα της εξαδέλφης, που *δεν* είναι Δωδεκαθεΐστρια, και η μητέρα της Μαιρούλας, που _λέει_ ότι είναι (η Μαιρούλα εννοώ, όχι η μητέρα της).

Α, και η είσοδος του Φέισμπουκ στο χρηματιστήριο.



ονέι μέιλ

Πρόκειται για δύο σαφώς διαφορετικά θέματα, που συναντιούνται, όμως, στον εσώτερο κύκλο των ερευνητικών μου ενδιαφερόντων.

Από τη μία διαβάζω πως οι ενεργές ουσίες του κουρκουμά ή κιτρινόριζας (αγγλιστί: turmeric) είναι πολλά υποσχόμενες για πλήθος εκφυλιστικών ασθενειών όπως είναι η αρθρίτιδα, η νόσος του Αλτσχάιμερ και ο καρκίνος του παχέος εντέρου. Αυτή τη στιγμή βρίσκονται στα τελικά τους στάδια 61 διαφορετικές κλινικές έρευνες, διαβάζω στην Βικιπαίδεια, και για να το γράφουν κάτι θα ξέρουν.

Από την άλλη ακούω πως η υπερκάλυψη της αρχικής δημόσιας προσφοράς μετοχών του Φέισμπουκ αναμένεται να είναι πολλαπλή -- κάτι πολλά υποσχόμενο για πλήθος άλλων προβλημάτων, κυρίως οικονομικών.

Προ ημερών, την ημέρα των (πρώτων) εκλογών, είχαμε μαζευτεί όλοι στο σπίτι: η Εξαδέλφη, η μητέρας της που είναι Ινδουίστρια, η Κυρία Καίτη που δεν είναι, και η Μαιρούλα, που μας είχε πει ότι είναι Δωδεκαθεΐστρια -- αλλά μπορεί και να μην είναι. Οι μισοί είχαμε κάτσει μπροστά από την τηλεόραση και περιμέναμε τα έξιτ πολ· οι άλλοι μισοί διαβάζαμε τις προβλέψεις για τις Γαλλικές εκλογές· και οι υπόλοιποι είχαμε ανοίξει το ίντερνετ και διαβάζαμε το μακρύ και το κοντό του καθενός.

Στο σκάιπ, τέλος, είχαμε τον ανιψιό, εκ Λονδίνου αυτός, που μας μετέδιδε το κλίμα εκεί. (Με δυό λόγια: κρύο, βροχή, πλήρης αδιαφορία.)

Πιάσαμε που λέτε την κουβέντα --με τον ανιψιό εννοώ-- και κάποια στιγμή μάς λέει για το Φέισμπουκ και για την επερχόμενη είσοδό του στο Χρηματιστήριο.

Βεβαίως δεν εννοούσε αυτό των Αθηνών. Το ενδεχόμενο να δούμε μετοχές Φέισμπουκ στην Σοφοκλέους παραμένει σχετικώς μακρυνό. Με όλα τούτα που συμβαίνουν δεν θα ήταν κακή ιδέα, βέβαια, αλλά άντε να πείσεις τον Ζούπερμαν, Ζούκερμαν, πώς τον λένε, να κατηφορήσει προς τα μέρη μας.

Εκατό δισεκατομμυριάκια δολλάρια φιλοδοξεί να συλλέξει ο νεαρός για την εταιρεία του και, την στιγμή που γράφονται αυτές εδώ οι γραμμές, όλα δείχνουν ότι θα το καταφέρει.

Βέβαια οι καιροί είναι δύσκολοι, και εκατό δις σήμερα τα έχεις και αύριο δεν τα έχεις.

Αλλά θα συμφωνήσετε, βέβαια, ότι είναι προτιμότερο να τα έχεις, έστω και αν είναι πρόσκαιρο.

Ο ανιψιός είχε μεγαλόπνοα σχέδια σχετικά με το Φέισμπουκ και τις μετοχές του, και με το πόσο θα ανέβουν με το που μπούνε στο Χρηματιστήριο και διάφορα άλλα συναφή, αλλά τον προσγειώσαμε απότομα.

Το Φέισμπουκ και οι μετοχές του μάς ενδιαφέρουν όλους, και όχι μόνο από καθαρά ακαδημαϊκής πλευράς, αλλά αυτή τη στιγμή προέχουν άλλα.

Όπως, για παράδειγμα, η αδυναμία να βγει Κυβέρνηση. Και το ότι πάμε ξανά σε εκλογές. (Σ.τ.Ε. ...ή πήγαμε ήδη. Τελικά, όλα θέμα τάιμινγκ είναι.) Και το απίστευτο τούτο φαινόμενο να βλέπεις το 7% του Ελληνικού λαού (ή έστω όσων αποφάσισαν να πάνε να ψηφίσουν) να εκφράζει την αγανάκτησή του μέσω των Χρυσών Αβγών.

Και βεβαίως το Κοτόπουλο Τζαλφρέζι (ΤΜ), που έχει αρχίσει ήδη να ετοιμάζεται στην κουζίνα.

Αυτό και μόνο μου δίνει τη δύναμη που χρειάζομαι για να συνεχίσω.

Σας ασπάζομαι,
Θείος Ακάκιος

Εσείς τι δικαιολογία έχετε που η διαφήμισή σας απουσιάζει από εδώ;

Τολμήστε να μας την πείτε από το talk2us@deltahacker.gr. Όπως και να 'χει θα επικοινωνήσουμε κι εμείς μαζί σας - ή έστω με τον ανταγωνιστή σας ;)

Προς το παρόν είμαστε απασχολημένοι με τη δημιουργία του καλύτερου τεχνολογικού περιοδικού με θεματολογία ethical hacking. Ξέρετε. Για ανήσυχους, δημιουργικούς ανθρώπους που δεν βλέπουν τον υπολογιστή τους ως τοστιέρα. Σε πολλούς εξ αυτών αρέσει και το The Big Bang Theory. Και το Fringe, φυσικά.



Skill: Beginner

Tags: mobile user, master.passwd, brute force, john the ripper

Κινδυνεύει το iPhone?

by Dr@G

Για ακόμη μία φορά θα φτάσουμε στα άδυτα του iOS: Θα εξετάσουμε διάφορα παράξενα χαρακτηριστικά του συστήματος που σχετίζονται με τους λογαριασμούς των χρηστών και θα δούμε πόσο επικίνδυνη μπορεί να αποδειχθεί η φυσική επαφή μ' ένα jailbroken iPhone. Επιπρόσθετα, με τη βοήθεια του john the ripper θα κάνουμε μια επίθεση ωμής βίας, με σκοπό να βρούμε το password του χρήστη root!

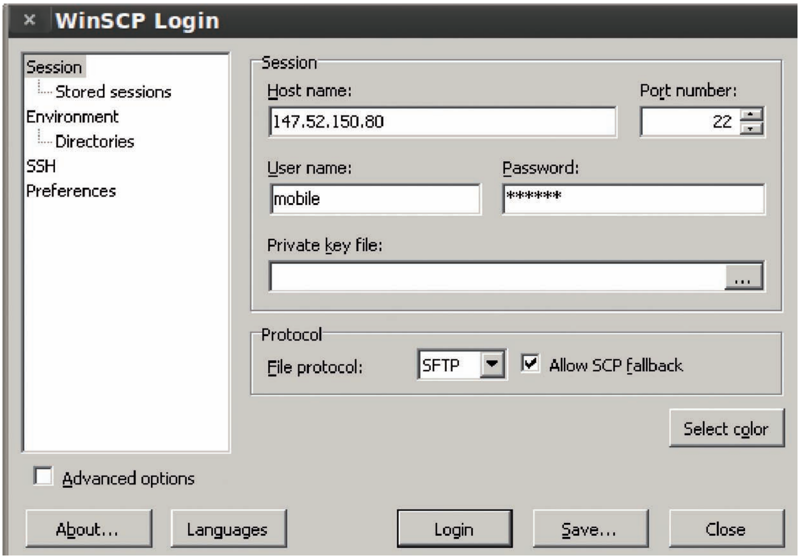
Η ιδέα για τη συγγραφή του άρθρου προέκυψε στο φετινό συνέδριο ασφάλειας AthCon. Εκεί γνώρισα φίλους που μέχρι τότε ήξερα μόνο από τη fan page του deltaHacker, ενώ συνάντησα και πολλούς παλιούς. Σε κάποιο διάλειμμα, πάνω σε μια κουβέντα για διάφορα θέματα του περιοδικού, αναφερθήκαμε στο άρθρο iOS vs iOS, που είχε δημοσιευτεί στο deltaHacker 005. Ακριβώς τότε, ένας από τους καλούς μας φίλους μου έδωσε το τηλέφωνό του, ένα iPhone 4S, και με προκάλεσε να "μπω στο κινητό". Προφανώς, είχε φροντίσει να αλλάξει το default root password και ήθελε να δει αν υπάρχουν άλλες τρύπες. Ωστόσο οι ζωντανές επιδείξεις δεν είναι στο DNA μου κι ενώ μπορούσα να τα καταφέρω, δεν το έκανα. Σκέφτηκα όμως ότι θα μπορούσα να δείξω τον τρόπο σε όλους τους αναγνώστες του deltaHacker, μέσα από ένα άρθρο: είναι αυτό που τώρα διαβάζετε. Έτσι, στη συνέχεια θα παρουσιάσω μερικούς τρόπους για την απόκτηση μη εξουσιοδοτημένης πρόσβασης σ' ένα jailbroken iPhone, χωρίς να έχω το password του root!

Ξεχασμένος χρήστης

Μετά το AthCon, ο φίλος με το iPhone 4S επέστρεψε στο σπίτι του, κάπου στη βόρεια Ελλάδα. Εκεί η ζωή του κυλούσε ομαλά και όμορφα, αλλά εγώ δεν μπορούσα να ησυχάσω. Ήθελα να τον ενημερώσω για τον κίνδυνο που διέτρεχε το κινητό του, οι προσωπικές φωτογραφίες, τα βίντεο κ.λπ. Έτσι, σκέφτηκα να του παρουσιάσω τους κινδύνους από μακριά! Μιλώντας στο chat του Facebook, τον ρώτησα αν πιστεύει ότι *δεν* μπορώ να μπω στο κινητό του. Η απάντηση του ήταν καταφατική. Τον προέτρεψα, λοιπόν, να προωθήσει το port 22 του router του στο jailbroken iPhone και να μου δώσει τη δημόσια IP του router. Ο καλός φίλος δέχθηκε, αλλά με προειδοποίησε ότι είχε εγκατεστημένο το εργαλείο fail2ban, για να κόβει τις επιθέσεις brute force. Επίσης, μου έδωσε μόνο πέντε λεπτά προθεσμία για να ολοκληρώσω το έργο μου! Όπως αντιλαμβάνεστε, είχα κι εγώ τα μυστικά μου και δεν υπήρχε κανένας λόγος να αρνηθώ τους όρους του. Μετά από λίγα λεπτά πραγματοποίησε το port forwarding, μου έδωσε την IP του (147.52.*.*) και με ειδοποίησε ότι ο χρόνος μετρούσε ήδη!

Δουλεύοντας σε ένα Laptop με Windows XP, άνοιξα αμέσως το WinSCP (http://winscp.net/eng/download.php)[1]. Προφανώς, ο φίλος μου είχε αλλάξει τον προεπιλεγμένο κωδικό πρόσβασης του χρήστη root, που είναι alpine, οπότε δεν είχε κανένα νόημα να τον δοκιμάσω. Έτσι, στράφηκα σε έναν άλλο χρήστη, με επίσης γνωστό password! Πρόκειται για το χρήστη mobile, ο οποίος δεν μπορεί να πραγματοποιήσει καμία αλλαγή στο σύστημα του κινητού (δεν έχει δικαιώματα διαχειριστή), αλλά μπορεί και βλέπει τα πάντα! Ο χρήστης mobile έχει πρόσβαση στις φωτογραφίες, στα βίντεο, στις ηχογραφήσεις, στις αποθηκευμένες διευθύνσεις, στα μηνύματα και πάει λέγοντας. Ο εν λόγω χρήστης έχει πρόσβαση σε κάθε είδους πληροφορία και μάλιστα σε χρόνο dt. Αναρωτιέστε ποιο είναι το συνθηματικό του; Μα, φυσικά, το alpine! Είναι παράξενο, αλλά η Apple έδωσε το ίδιο password στον χρήστη root και στον χρήστη mobile. Σκεφτείτε τώρα το εξής: Από τη μία, οι απλοί χρήστες κάνουν jailbreak και δεν αλλάζουν το password του root, διότι δεν γνωρίζουν την ύπαρξή του -- ούτε τους σχετικούς κινδύνους. Από την άλλη, οι προχωρημένοι χρήστες αλλάζουν (σχεδόν) πάντα το password του root, αλλά οι περισσότεροι αγνοούν την ύπαρξη του χρήστη mobile. Επομένως η πιθανότητα να μπει κανείς σε ένα jailbroken iPhone είναι πάρα πολύ μεγάλη! Ας επιστρέψουμε όμως στην ιστορία με το φιλαράκι από το Βορρά...

1. Θα μπορούσα να είχα χρησιμοποιήσει και το PuTTY.



Εικόνα 1
Σύνδεση στο iPhone με το WindSCP...

Αφού άνοιξα το WinSCP, συμπλήρωσα στα κατάλληλα πεδία τη διεύθυνση IP του στόχου μου (147.52.*.*), το username (mobile) και το password (alpine) (βλ. εικόνα 1). Έτσι, συνδέθηκα στο κινητό και ήμουν έτοιμος να δείξω στο φίλο τους κινδύνους που του είχα περιγράψει. Καταρχάς, από το GUI του WinSCP είδα αμέσως ότι ο χρήστης mobile είναι ιδιοκτήτης (owner) όλων των γαργαλιστικών καταλόγων και άρα έχει πλήρη πρόσβαση στο περιεχόμενό τους (εικόνα 2). Παρεμπιπτόντως, οι φάκελοι με το μεγαλύτερο ενδιαφέρον είναι οι Media, Library, Downloads, Documents και Applications. Η αντιγραφή αρχείων ή και ολόκληρων καταλόγων με το WinSCP είναι πανεύκολη. Έτσι, η επόμενη κίνησή μου ήταν να μεταφερθώ στο φάκελο /private/var/mobile/Media/DCIM/100APPLE. Εκεί βρίσκονται όλες οι φωτογραφίες και τα βίντεο του κατόχου του κινητού! Όπως αντιλαμβάνεστε, δεν κάθισα να το φιλοσοφήσω. Πήρα δύο φωτογραφίες στην τύχη και αποσυνδέθηκα. Πλέον, διέθετα κάθε απόδειξη όχι μόνο για το ότι είχα μπει στο κινητό, αλλά και για το ότι είχα πρόσβαση στα προσωπικά του δεδομένα. Φυσικά, τα παραπάνω θα μπορούσα να τα είχε πετύχει εξίσου εύκολα και από τη γραμμή εντολών του Linux. Εκεί, θα μπορούσα να είχε δώσει κάτι τέτοιο:

```
$ scp mobile@147.52.150.80:/private/var/mobile/Media/
DCIM/100APPLE/ \{IMG_0633.JPG,IMG_0634.JPG\} ~/
```

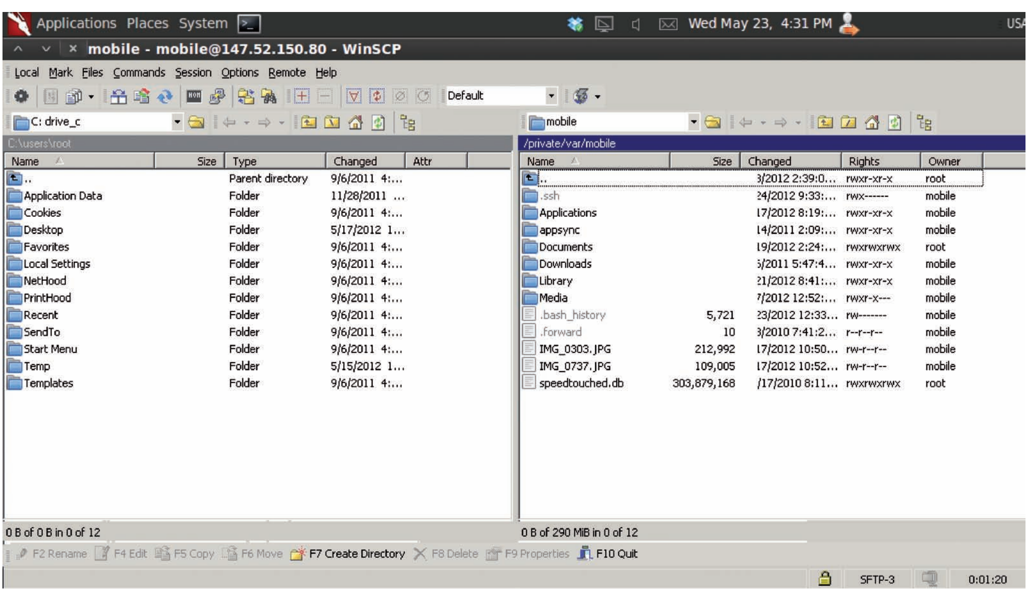
Είδατε πόσο εύκολο ήταν; Λίγα λεπτά μετά, ζήτησα από το φίλο μου να μπει στο Skype, για να του δείξω κάτι. Όταν συνδέθηκε δεν του είπα κουβέντα. Το μόνο που έκανα ήταν να σύρω τις δύο φωτογραφίες από το κινητό του, στο παράθυρο συνομιλίας ;) Το φιλαράκι μου κοκάλωσε, ενώ λίγο μετά έβαλε τις φωνές κι άρχισε να με ρωτάει πώς το έκανα. Τον συμβούλεψα να κοιτάξει το αρχείο /etc/master.passwd. Το συγκεκριμένο αρχείο παρουσίαζε το ακόλουθο περιεχόμενο:

```
#
# User Database
#
```

```
# This file is the authoritative user database.
##
nobody:*:-2:-2::0:0:Unprivileged User:/var/empty:/usr/bin/false
root:OGR8L5s.paOrc:0:0::0:0:System Administrator:/var/root:/bin/sh
mobile:VMxOANLTyhUTQ:501:501::0:0:Mobile User:/var/mobile:/bin/sh
daemon:*:1:1::0:0:System Services:/var/root:/usr/bin/false
_wireless:*:25:25::0:0:Wireless Services:/var/wireless:/usr/bin/false
_securityd:*:64:64::0:0:securityd:/var/empty:/usr/bin/false
_mdnsresponder:*:65:65::0:0:mDNSResponder:/var/empty:/usr/bin/false
_sshd:*:75:75::0:0:sshd Privilege separation:/var/empty:/usr/bin/false
_unknown:*:99:99::0:0:Unknown User:/var/empty:/usr/bin/false
```

Εδώ παρατηρούμε ξεκάθαρα ότι υπάρχουν δύο χρήστες στο κινητό: Ο διαχειριστής root και ο απλός χρήστης mobile. Οι συμβολοσειρές "OGR8L5s.paOrc" και "VMxOANLTyhUTQ" αποτελούν τις κρυπτογραφημένες εκδοχές των κωδικών των δύο χρηστών. Η κρυπτογράφηση έχει γίνει κατά DES (Data Encryption Standard) και οι δύο πρώτοι χαρακτήρες σε κάθε συμβολοσειρά αποτελούν το λεγόμενο salt. Για παράδειγμα, στην περίπτωση του χρήστη mobile με κρυπτογραφημένο κωδικό το VMxOANLTyhUTQ, το salt είναι το "VM". Στον αλγόριθμο DES, το "αλατάκι" συμμετέχει στη διαδικασία κρυπτογράφησης και το τέλος ενσωματώνεται αυτούσιο στην αρχή του κρυπτογραφημένου string. Παρακάτω φαίνεται ένα μικρό πρόγραμμα σε Python, το οποίο δέχεται ένα συνθηματικό κι ένα salt και πραγματοποιεί την κρυπτογράφηση κατά DES:

```
#!/usr/bin/python
# DES_Crypter.py
import crypt
```



Εικόνα 2
Τα αρχεία του χρήστη mobile είναι στον φάκελο / private/var/mobile.

```
passwd=raw_input('Password: ')
salt=raw_input('Salt: ')
encryptedpass=crypt.crypt(passwd,salt)
print encryptedpass
```

Σημειώστε ότι η βιβλιοθήκη crypt λειτουργεί μόνο σε συστήματα Unix/Linux. Επομένως, το πρόγραμμά μας τρέχει άνετα και σε MAC ή iPhone. Δείτε μια δοκιμαστική εκτέλεση:

```
root@bt:~# python DES_Crypter.py
Password: alpine
Salt: VM
VMxOANLTyhUTQ
```

Δώσαμε το alpine και πήραμε ακριβώς το ίδιο string, το οποίο συναντήσαμε και στο αρχείο master.passwd, ως συνθηματικό του χρήστη mobile! Τώρα, μπορεί να νομίζετε ότι αυτό το πρόγραμμα είναι άχρηστο και ότι δεν το έχουμε καμία ανάγκη. Σε λίγο θα σας αποδείξουμε το αντίθετο...

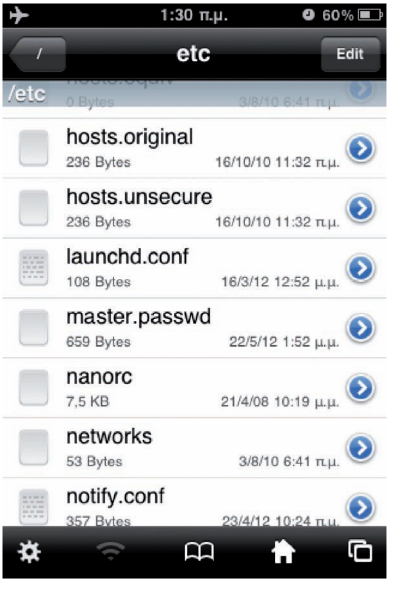
Φυσική επαφή

Ας γυρίσουμε το χρόνο στο AthCon 2012 και συγκεκριμένα στη στιγμή που ο φίλος μας μου έδωσε το iPhone του. Θα μπορούσα άραγε να γίνω System Administrator της συσκευής, χωρίς να ξέρω το password του root; Υπάρχει τρόπος να μαντέψουμε το password του root; Ας υποθέσουμε ότι αποδεχόμουν την πρόκληση του φίλου.

Να σημειωθεί εδώ ότι διαθέτω κι εγώ jailbroken iPhone (το 3GS). Στο κινητό έχω εγκατεστημένη τη γλώσσα Python κι έχω περασμένο το μικρό πρόγραμμα που είδαμε λίγο πριν (το DES_Crypter.py). Μόλις παίρνω το iPhone του φίλου μας, του απαντώ ευγενικά ότι αποδέχομαι την πρόκληση μόνο και μόνο για χάρη της επιστημονικής έρευνας ;) Του ζητώ μόνο πέντε με δέκα λεπτά ηρεμίας κι απομόνωσης. Εδώ που τα λέμε, ο χρόνος εξαρτάται από το πόσο προετοιμασμένος είναι κανείς, όπως κι από τα προγράμματα που έχει εγκαταστήσει ο φίλος στο κινητό του. Φυσικά εγώ είμαι καλά προετοιμασμένος, αλλά δεν έχω ιδέα για το τι θα συναντήσω στο ξένο iPhone.

Αφού μείνω μόνος, ξεκινώ το θεάρεστο έργο μου. Για να δω τα αρχεία του τηλεφώνου ως διαχειριστής, χωρίς να γνωρίζω κάποιο password, χρειάζομαι το iFile. Ο φίλος μου δεν το έχει εγκατεστημένο και χρειάζεται να σπαταλήσω λίγα λεπτά για να το βρω στο Cydia και να το προσθέσω. Αμέσως μετά το τρέχω και πηγαίνω στον φάκελο /etc. Εκεί βρίσκω το αρχείο master.passwd και χωρίς να χάνω χρόνο το ανοίγω με τον Text Viewer (εικόνα 3). Στο πάνω αριστερό άκρο του Viewer εμφανίζεται ένα κουμπάκι, το οποίο μας επιτρέπει να επεμβαίνουμε σε κάποιο αρχείο (edit), ακόμη κι όταν δεν έχουμε δικαιώματα διαχειριστή (εικόνα 4). Όπως φαντάζεστε, εγώ πατάω αμέσως στο edit. Στη συνέχεια, πριν κάνω οποιαδήποτε επέμβαση, αντιγράφω τους κρυπτογραφημένους κωδικούς του root και του mobile στο δικό μου κινητό. Αυτό γίνεται για λόγους που θα φανούν αργό-

Εικόνα 3
Τα περιεχόμενα του καταλόγου /etc. Ανάμεσα στα διάφορα αρχεία, εκείνο με το μεγαλύτερο ενδιαφέρον είναι το master.passwd!





Εικόνα 4
Το περίφημο αρχείο master.passwd, με τους λογαριασμούς των χρηστών και τα κρυπτογραφημένα κατά DES passwords.

τερα. Μετά τρέχω το προγραμματάκι μου σε Python, το οποίο πραγματοποιεί κρυπτογραφήσεις κατά DES:

```
drg-iphone:~ root# python DES_crypter.py
Password: hackedbydrg
Salt: bb
bbSIhgBT84cRc
drg-iphone:~ root#
```

Όπως βλέπετε, στο πρόγραμμα δίνω έναν κωδικό τον οποίο μπορώ να θυμάμαι εύκολα κι ένα τυχαίο salt. Το προϊόν της κρυπτογράφησης το σημειώνω προσεκτικά, για να το μεταφέρω στο iPhone του φίλου μου. Συγκεκριμένα, το γράφω στη θέση του κρυπτογραφημένου κωδικού του root! Η σχετική γραμμή στο αρχείο master.passwd έχει πάντα την ακόλουθη μορφή:

```
root:[Κρυπτογραφημένος κωδικός]:0:0::0:0:System
Administrator:/var/root:/bin/sh.
```

Στο κινητό του φίλου μου, λοιπόν, βρίσκω την παρακάτω γραμμή:

```
root:zFfX0oEn1TogE:0:0::0:0:System Administrator:/
var/root:/bin/sh
```

Την αλλάζω σε

```
root:bbIqeSbsCDeUM:0:0::0:0:System Administrator:/var/root:/bin/sh
```

Μετά τη μικρή αυτή επέμβαση πατάω Save και Done (εικόνα 5). Το έργο μου είχε ολοκληρωθεί: Μπορεί να μην έχω μαντέψει το password του root, αλλά το έχω αντικαταστήσει με κάποιο άλλο, το οποίο και γνωρίζω. Έτσι, μπορώ ανά πάσα στιγμή να συνδεθώ ως διαχειριστής! Μάλιστα, για να βεβαιωθώ (και μεταξύ μας, για να χαρώ και λίγο) για το κατόρθωμά μου, ανοίγω το mobile-terminal στο κινητό του φίλου και κάνω τα ακόλουθα:

```
x_factor-iphone:~ mobile$ su -
Password: hackedbydrg        /*δεν εμφανίζεται στην οθόνη*/
X_factor-iphone:~ root#
```

Όπως βλέπετε, το su εκτελείται κι ολοκληρώνεται κανονικά. Το νέο password είχε ενεργοποιηθεί και μπορώ να γίνω root κανονικά! Σ' αυτό το στάδιο έχω αντιμετωπίσει την πρόκληση του φίλου μας με πλήρη επιτυχία. Ωστόσο, δεν μου είναι αρκετό αυτό. Θέλω να κάνω ακόμα πιο εντυπωσιακή την επέμβαση, βρίσκοντας το αρχικό password του root. Ευτυχώς που έχω σημειώσει την κρυπτογραφημένη του μορφή, πριν την αντικαταστήσω.

Αποκρυπτογράφηση

Για αυτή την εργασία καταφεύγω στο laptop και στην αγαπημένη μου διανομή Linux, το Backtrack. Για να διευκολύνω τη διαδικασία της αποκρυπτογράφησης, μεταφέρω στον υπολογιστή μου ολόκληρο το αρχείο master.passwd, από το iPhone. Δίνω για την ακρίβεια κάτι τέτοιο:

```
root@bt:~# scp root@192.168.1.9:/etc/master.passwd ~/pass.txt
```

(το IP του iPhone ήταν 192.168.1.9). Έτσι, το αρχείο αντιγράφεται και ταυτόχρονα μετονομάζεται σε pass.txt. Στη συνέχεια το ανοίγω με έναν text editor, για να το επαναφέρω στην αρχική του μορφή. Ουσιαστικά, αντικαταθιστώ τον δικό μου, κρυπτογραφημένο κωδικό, με τον κρυπτογραφημένο κωδικό του φίλου μας. Παρακάτω φαίνεται η τελική μορφή του master.passwd. (Εδώ που τα λέμε, ακριβώς έτσι το συναντήσαμε στο iPhone του φίλου, την πρώτη φορά που το ανοίξαμε...)

```
#
# User Database
#
# This file is the authoritative user database.
##
nobody:*:-2:-2::0:0:Unprivileged User:/var/empty:/usr/bin/false
root:zFfX0oEn1TogE:0:0::0:0:System Administrator:/var/root:/bin/sh
mobile:VMxOANLTyhUTQ:501:501::0:0:Mobile User:/var/mobile:/bin/sh
daemon:*:1:1::0:0:System Services:/var/root:/usr/bin/false
_wireless:*:25:25::0:0:Wireless Services:/var/wireless:/usr/bin/false
_securityd:*:64:64::0:0:securityd:/var/empty:/usr/bin/false
_mdnsresponder:*:65:65::0:0:mDNSResponder:/var/empty:/usr/bin/false
_sshd:*:75:75::0:0:sshd Privilege separation:/var/empty:/usr/bin/false
_unknown:*:99:99::0:0:Unknown User:/var/empty:/usr/bin/false
```

Στη συνέχεια επιστρατεύω το john the ripper, το super εργαλείο "σπασίματος" κρυπτογραφημένων κωδικών. Σημειώστε ότι διατίθεται και για iPhone, μέσω του Cydia. Το john the ripper βρίσκεται στη διαδρομή /pentest/passwords/john/. Έτσι, στη γραμμή εντολών του backtrack, δίνω τα ακόλουθα:

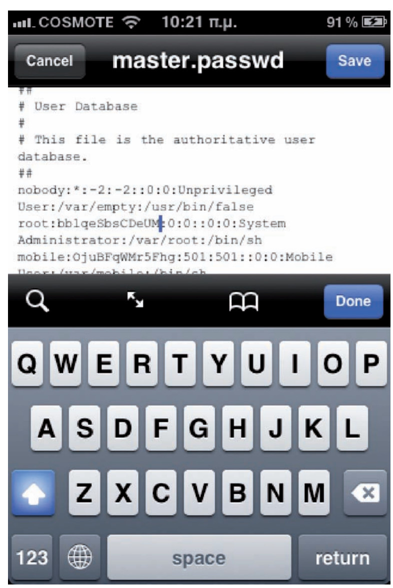
```
root@bt:~# cd /pentest/passwords/john
root@bt:/pentest/passwords/john# ./john ~/pass.txt
```

Τα αποτελέσματα φαίνονται στην εικόνα 6. Ο κωδικός που χρησιμοποιούσε ο φίλος μας για το χρήστη root, ήταν η λέξη hacker. Στο χρήστη mobile δεν είχε κάνει καμία αλλαγή, όπως άλλωστε και οι περισσότεροι ιδιοκτήτες jailbroken iPhone!

Προστασία

Συνήθως, οι κωδικοί που χρησιμοποιούν οι χρήστες στα κινητά τηλέφωνα αποτελούνται από λίγους χαρακτήρες. Με έναν καλό επεξεργαστή, ενδεχομένως και με μια ή δύο κάρτες γραφικών και φυσικά με ένα καλό wordlist, μπορούμε να σπάσουμε και τους πιο δύσκολους κωδικούς! Επομένως, για να μην πέφτουμε θύματα της άγνοιας ή της υπερβολικής αυτοπεποίθησης, καλό θα ήταν να ορίζουμε ένα πραγματικά δύσκολο συνθηματικό για τον root και το ίδιο να κάνουμε με τον χρήστη mobile! Κατά τα γνωστά, αυτά τα δύο μέτρα μπορούν να ληφθούν πανεύκολα από τη γραμμή εντολών. Αρκεί να

Εικόνα 5
Η αντικατάσταση του κρυπτογραφημένου password του root, με το δικό μας κρυπτογραφημένο password. Με αυτόν τον τρόπο ο χρήστης root θα προστατεύεται πλέον από έναν κωδικό που ορίσαμε εμείς!

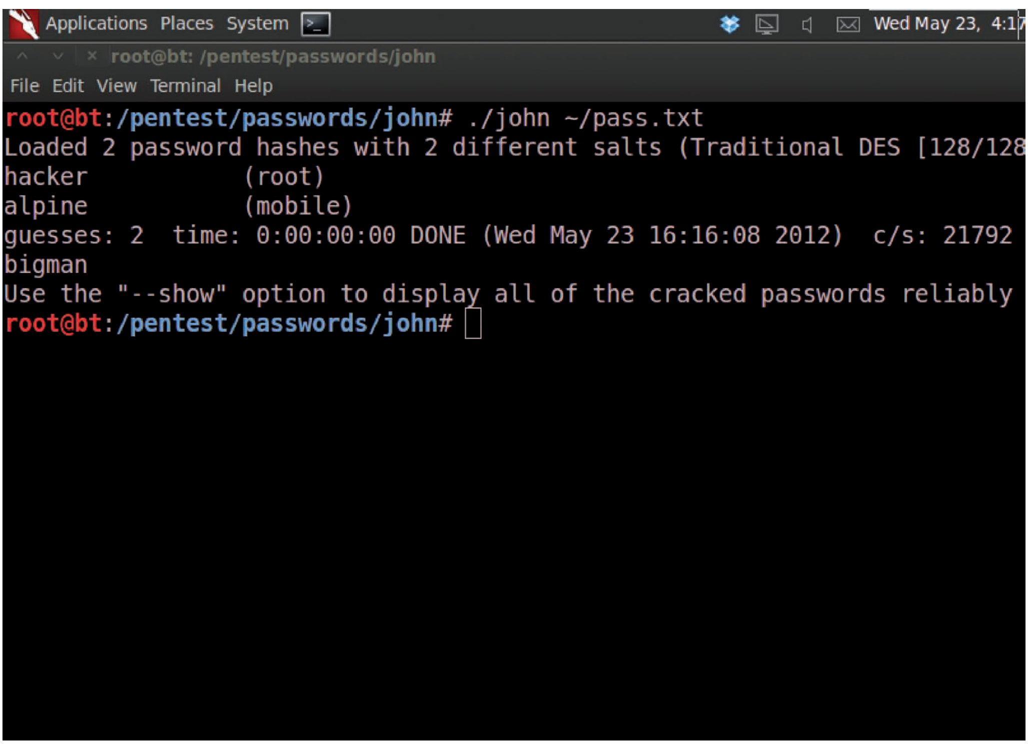


συνδεθούμε στο κινητό με έναν SSH client ή με την εφαρμογή mobile-terminal, και να χρησιμοποιήσουμε το προγραμματάκι passwd. Παρακάτω φαίνεται η αλλαγή του κωδικού για τον χρήστη mobile:

```
login as: mobile
mobile@192.168.1.9's password: alpine /*δεν εμφανίζεται στην οθόνη*/
drg-iphone:~ mobile$ passwd
Changing password for mobile.
Old password: alpine                        /*δεν εμφανίζεται στην οθόνη*/
New password: νέος_ισχυρός_κωδικός          /*δεν εμφανίζεται στην οθόνη*/
Retype new password: νέος_ισχυρός_κωδικός   /*δεν εμφανίζεται στην οθόνη*/
drg-iphone:~ mobile$
```

Τέλος, θα ήταν φρόνιμο να μη δίνουμε το κινητό μας τηλέφωνο σε "αθώους" αγνώστους, ούτε και σε "αθώους" φίλους ;) Υπερβολικό, θα μου πείτε, αλλά είναι καλό να προσέχουμε πάντα το κινητό μας και να μην ξεφεύγει ποτέ από την προσοχή μας.

Τώρα, αν όλα αυτά σας φάνηκαν απλά και ακίνδυνα, ετοιμαστείτε: Στο επόμενο τεύχος θα εξετάσουμε την εισαγωγή κώδικα σε *μη* jailbroken iPhone! (Σ.τ.Ε. Ανυπομονούμε!)



Εικόνα 6
Με τη χρήση του john the ripper καταφέραμε να σπάσουμε όλα τα κρυπτογραφημένα συνθηματικά! Το password του χρήστη root ήταν "hacker". Ο ταπεινός χρήστης mobile δεν είχε αλλάξει ποτέ password...



Skill: Advanced

Tags: databases, tables, users, members

Ο πράκτοράς μας συνεχίζει να κάνει όσα του ζητάνε, χωρίς δισταγμό! Πρέπει να κερδίζει διαρκώς την εμπιστοσύνη των άλλων μελών της ομάδας WHITE, για να μην κινήσει την παραμικρή υποψία. Πριν καλά-καλά ολοκληρώσει την προηγούμενη αποστολή, του ανέθεσαν μια νέα: Την άλωση μιας database και το ψάρεμα πολύτιμων δεδομένων...

Ομάδα WHITE: Χτύπημα στη βάση!

by Thiseas

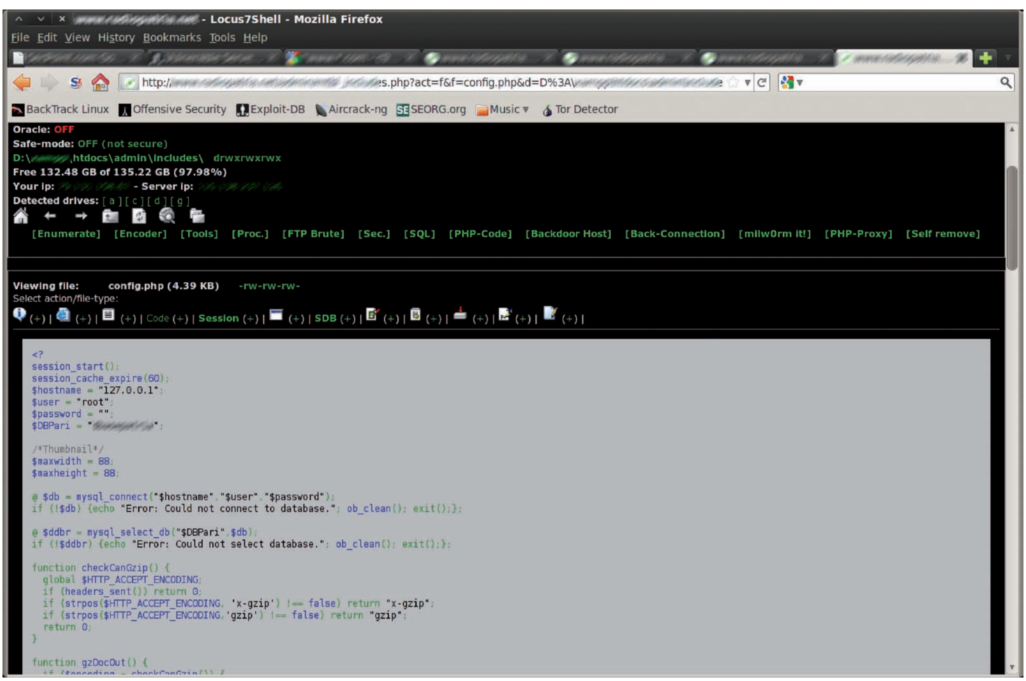
Μη νομίζετε ότι κάθε αποστολή της ομάδας WHITE αποδίδει εκατομμύρια στους εγκεφάλους και στα μέλη της. Ακριβώς γι' αυτό, η ομάδα δεν σταματά ποτέ. Οφείλει να εντοπίζει διαρκώς νέους στόχους, να προετοιμάζει τα επόμενα βήματά της και φυσικά να εξαντλεί κάθε περιθώριο κέρδους! Ο άνθρωπός μας έχει μπει για τα καλά στους ρυθμούς της ομάδας και δεν ησυχάζει ποτέ. Μόλις κατέλαβε μερικά μηχανήματα και τα μετέτρεψε σε jump-boxes (βλ. σχετικό άρθρο στο deltaHacker 008), του ανατέθηκε νέα αποστολή: Να αντλήσει όσο πιο πολλές πληροφορίες μπορούσε, από έναν ήδη κατειλημμένο server. Περιττό να σας πούμε ότι ο πράκτοράς μας τα κατάφερε και πάλι, μάλιστα χωρίς να στηριχτεί σε έτοιμα εργαλεία. Ο server ήταν "φαγωμένος" και προσέφερε ήδη ένα web-shell, αλλά για την πρόσβαση στη βάση χρειάστηκε απλή λογική, υπομονή και λίγη πονηριά ;)

Ας παρουσιάσουμε λοιπόν ως συνήθως την αυτούσια αναφορά του shadow[1], για το πώς κατέλαβαν την βάση δεδομένων του συγκεκριμένου server.

Πώς δουλεύω...

Πριν μιλήσω για την επίθεση στη βάση θέλω να περιγράψω τον τρόπο δράσης της ομάδας, όπως τον έχω βιώσει εγώ. Κάθε φορά που αποφασίζουν να επιτεθούν σε κάποιο server, θέτουν τους εξής στόχους:

1. **Κατάληψη συστήματος.** Ένας server έχει πάντα μεγάλη αξία. Ακόμα κι όταν δεν πρόκειται για μηχάνημα υψηλού ενδιαφέροντος, μπορεί να χρησιμεύσει ως jump-box. (Σ.τ.Ε. Για τα jump-boxes μπορείτε να διαβάσετε στο άρθρο που αρχίζει από τη σελίδα 56 του deltaHacker 008.)



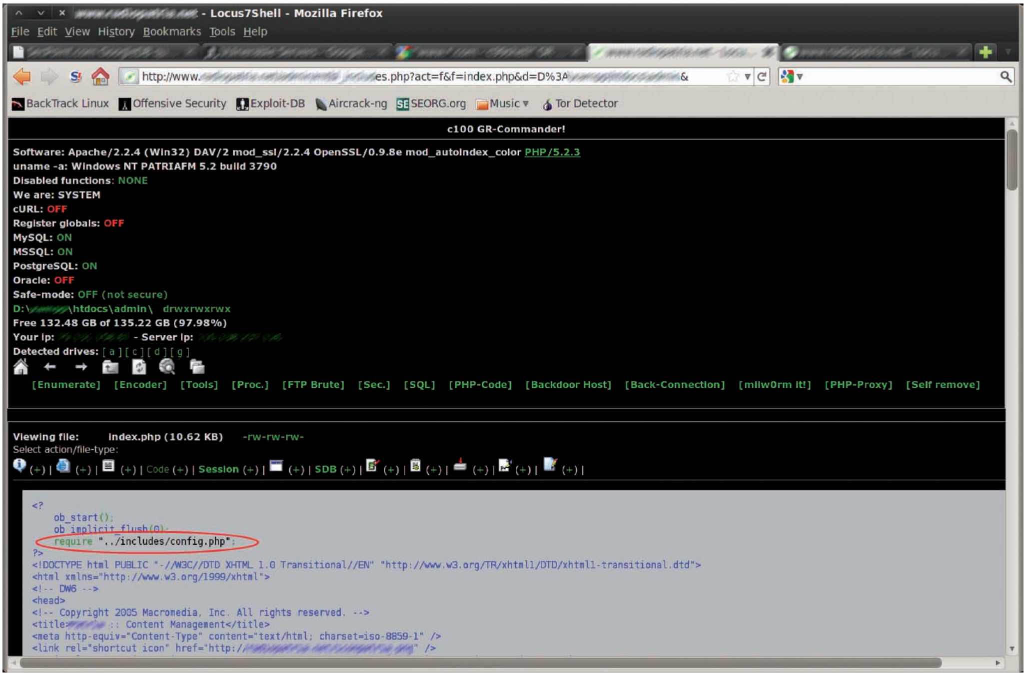
Εικόνα 1
Ορίστε τα credentials για τη σύνδεση στη βάση δεδομένων! Είναι δυνατόν να μην υπάρχει password για το χρήστη root; Αυτό θα ήταν πολύ ανόητο από την πλευρά του διαχειριστή...

1. Το shadow είναι το ψευδώνυμο του πράκτορά μας.

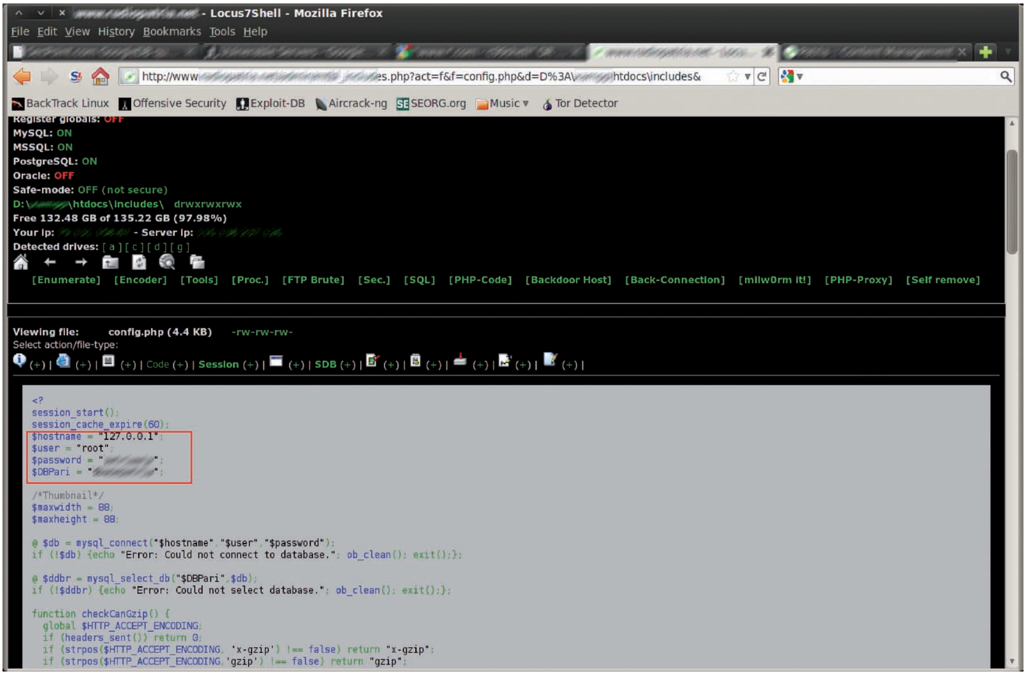
2. **Κατάληψη βάσης δεδομένων.** Μια βάση δεδομένων αποτελεί συνήθως πηγή για εξαιρετικά ενδιαφέρουσες πληροφορίες. Για την ομάδα WHITE, αυτές οι πληροφορίες αποτελούν πρώτης τάξης εμπόρευμα για τη μαύρη αγορά ή ένα αποτελεσματικό μέσο πίεσης, για τον εκβιασμό των ιδιοκτητών. Με λίγα λόγια, οι πληροφορίες είναι χρήμα!
3. **Κατάληψη λοιπών υπηρεσιών.** Σειρά έχουν ο mail κι ο FTP server. Αρκετοί διαχειριστές και πολλοί χρήστες, ειδικά εκείνοι των μεγάλων εταιριών, διακινούν πολύτιμες πληροφορίες μέσω αυτών των υπηρεσιών. Μάλιστα, χρησιμοποιώντας ασφαλείς συνδέσεις θεωρούν ότι είναι και οι ίδιοι απόλυτα ασφαλείς... Τι μύθος! Η ασφαλής σύνδεση σε προστατεύει μόνο από τους sniffers και όχι από αυτούς που έχουν ήδη καταλάβει τον server. Αν σου "φάνε" τον mail server ή τον secure FTP server, μαζί με όλα τα passwords, τότε η ασφαλής σύνδεση είναι άχρηστη. Παρεμπιπτόντως, αν θέλετε να είστε 100% ασφαλείς στέλνετε τα κείμενά σας σε κρυπτογραφημένη μορφή. Εγώ, ας πούμε, για να στείλω τις αναφορές μου χρησιμοποιώ το Axantum Encryption Software (www.axantum.com/AxCrypt).

Κατάληψη βάσης

Όταν αποκτάω πρόσβαση στο σύστημα αρχείων ενός στόχου, αρχίζω να μελετάω διάφορα scripts τα οποία ενδέχεται να με βοηθήσουν στις επιμέρους αποστολές μου. Αυτή τη φορά, έχοντας σαν στόχο την επίθεση στη βάση δεδομένων, ξεκίνησα το ψάξιμο από τα αρχεία της εφαρμογής web που χρησιμοποιεί τη βάση. Τα αγαπημένα μου αρχεία είναι τα config.php. Σχεδόν πάντα, μέσα σε αυτά τα αρχεία εμφανίζονται τα credentials



Εικόνα 2
Πολύ συχνά, το index.php αποτελεί την αφετηρία για την ανακάλυψη πολύτιμων πληροφοριών.



Εικόνα 3
Αυτή τη φορά τα credentials μοιάζουν αληθινά!

για την πρόσβαση στη βάση δεδομένων και μάλιστα χωρίς καμία κρυπτογράφηση. Έτσι, με φούρια θα έλεγε κανείς, μπήκα στον κατάλογο "...\httpdocs\admin\" κι άρχισα να ψάχνω εκεί μέσα. Μετά από λίγο μπήκα και στον κατάλογο includes. Εκεί, βρήκα αυτό που έψαχνα! Τα περιεχόμενα του config.php φαίνονται στην εικόνα 1.

Τα αρχεία "config.php" δεν έγιναν τυχαία τα αγαπημένα μου. Όταν άνοιξα το εν λόγω αρχείο στο κατειλημμένο μηχάνημα, βρήκα πολλά ενδιαφέροντα στοιχεία. Καταρχάς, η βάση είναι τοπική (hostname = "127.0.0.1") και το username για τη σύνδεση της εφαρμογής στη βάση, είναι "root". Πολύ καλά μέχρι εδώ. Όμως το password μοιάζει να είναι κενό (password=""). Είναι δυνατόν; Δοκίμασα να μπω στη βάση με username το root και με κενό password και, φυσικά, έφαγα πόρτα. Κάτι δεν πήγαινε καλά, αλλά δεν είχα ιδέα τι ήταν! Στην αρχή φοβήθηκα μήπως είχα πέσει σε honeypot (βλ. deltaHacker 006 και 007). Μετά όμως σκέφτηκα ότι μπορεί να είχα ανοίξει ένα αρχείο από παλιά εγκατάσταση της εφαρμογής web -- κάποιο αρχείο που δεν χρησιμοποιείται πλέον. Έτσι, αποφάσισα να συνεχίσω για λίγο το ψάξιμο. Μπορεί να είχα εγκαταλείψει το config.php, αλλά δεν θα εγκατέλειπα και όλα τα scripts. Το επόμενο αρχείο που μου φάνηκε ενδιαφέρον ήταν το "...\httpdocs\admin\index.php". Ως γνωστόν, το httpdocs αποτελεί το βασικό directory για σχεδόν όλες τις εφαρμογές web. Το admin νομίζω ότι είναι αυτονόητο! Όσο για το index.php, αυτό αποτελεί το βασικό αρχείο έναρξης για κάθε εφαρμογή γραμμένη σε PHP. Στην εικόνα 2 φαίνονται τα περιεχόμενα του εν λόγω αρχείου.

Εκεί μέσα παρατήρησα κάτι ενδιαφέρον: Το τμήμα της εφαρμογής web, το οποίο βρίσκεται στον κατάλογο admin, διαβάζει ένα διαφορετικό "config.php", σε σχέση με εκείνο που είχα ανοίξει προηγουμένως. Την είχα πατήσει σαν αρχάριος! Προηγουμένως είχα ανοίξει το "...\httpdocs\

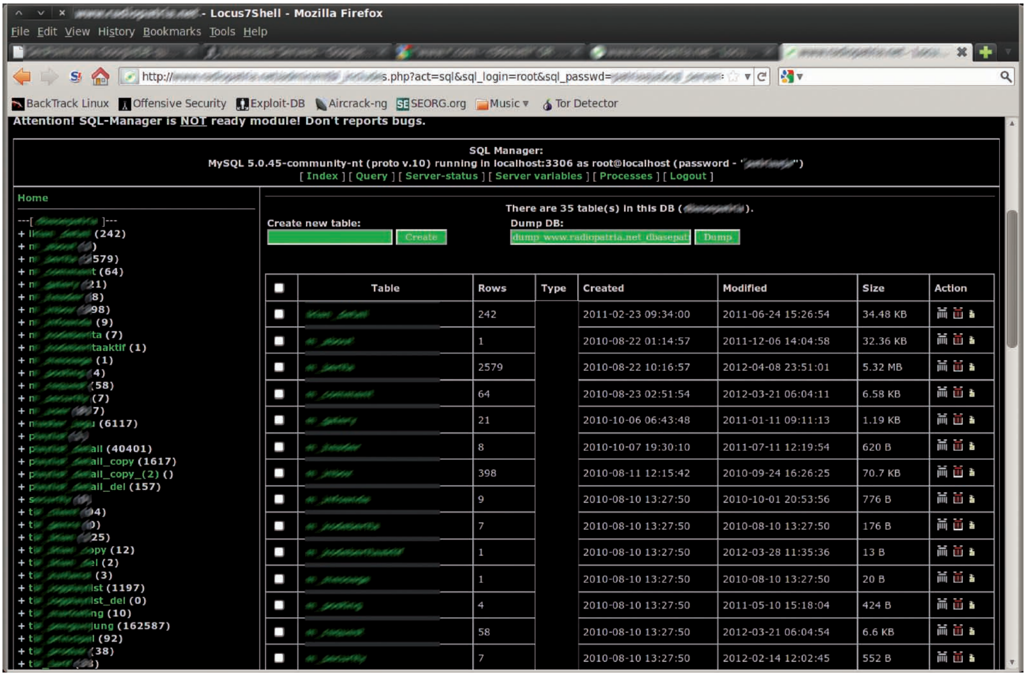
admin\includes\config.php", ενώ σύμφωνα με το index.php θα έπρεπε να είχα ανοίξει το "...\httpdocs\includes\config.php". Με ανυπομονησία πήγα στον κατάλογο και άνοιξα το σωστό "config.php". Στην εικόνα 3 μπορείτε να δείτε τα περιεχόμενα του αρχείου...

Εκεί τα credentials ήταν αληθοφανή. Αυτή τη φορά γνώριζα ότι η βάση είναι τοπική, ήξερα ότι το username είναι root και είχα στη διάθεσή μου το αντίστοιχο password, καθώς και το όνομα της βάσης. Το τελευταίο στοιχείο φαινόταν από την τιμή της μεταβλητής DBPari. Αναρωτιέστε πώς ήξερα ότι αυτή η μεταβλητή διατηρεί το όνομα της βάσης, την οποία χρησιμοποιεί η εφαρμογή; Αυτό ήταν απλό! Λίγες γραμμές χαμηλότερα, η συγκεκριμένη μεταβλητή δινόταν σαν παράμετρος στη συνάρτηση mysql_select_db(). Άψογα! Με όλα αυτά τα στοιχεία συγκεντρωμένα, δεν είχα παρά να δοκιμάσω μια σύνδεση στη βάση. Για το σκοπό αυτό χρησιμοποίησα το σχετικό link του c99 web-shell (είναι σημειωμένο στην εικόνα 3). Εκεί έδωσα τα credentials -- και κατάφερα να συνδεθώ με επιτυχία! Τα αποτελέσματα φαίνονται στην εικόνα 4.

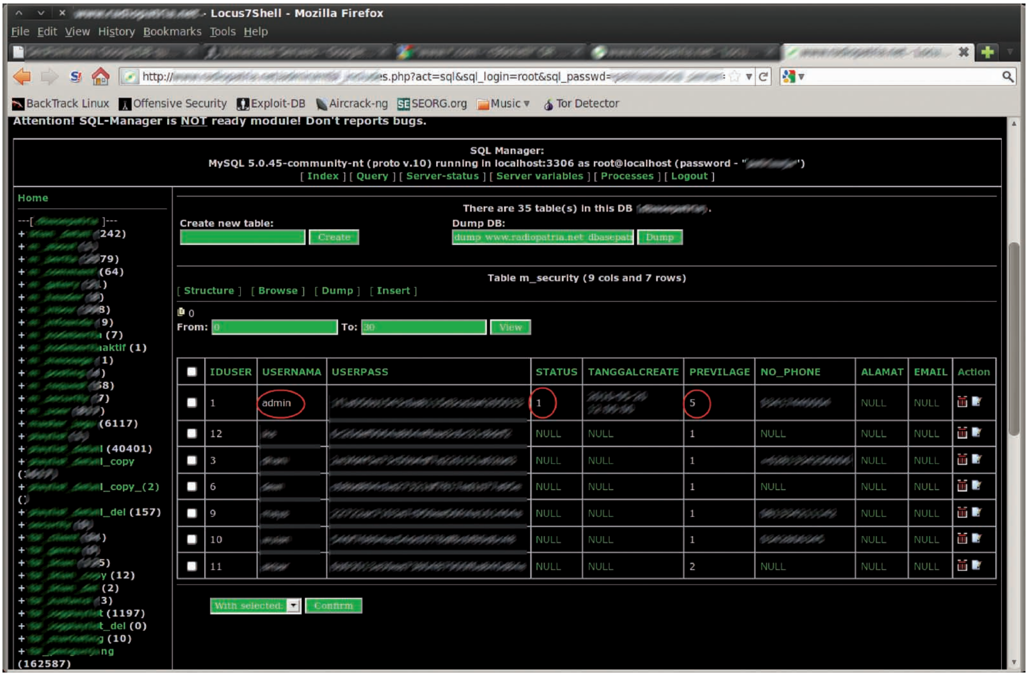
Πλέον, είχα μπροστά μου έναν μεγάλο κατάλογο με όλους τους πίνακες της βάσης, την ημερομηνία δημιουργίας του καθενός, το μέγεθος κ.λπ. Η πρώτη μου κίνηση ήταν να πάρω ένα αντίγραφο της βάσης (dump) στο δικό μου μηχάνημα. Ποτέ δεν ξέρεις τι γίνεται! Ανά πάσα στιγμή μπορεί να σε ανακαλύψουν και να σε πετάξουν έξω. Καλό θα ήταν, λοιπόν, να έχω πάρει όσα πιο πολλά στοιχεία μπορούσα...

Μέσα στη βάση

Ένας από τους πρώτους πίνακες που κοιτάζω όταν «τρώω» μια βάση δεδομένων, είναι ο πίνακας των χρηστών της εφαρμογής. Κάθε σοβαρή εφαρμογή που χρησιμοποιεί βάση δεδομένων, διατηρεί σε αυτήν έναν



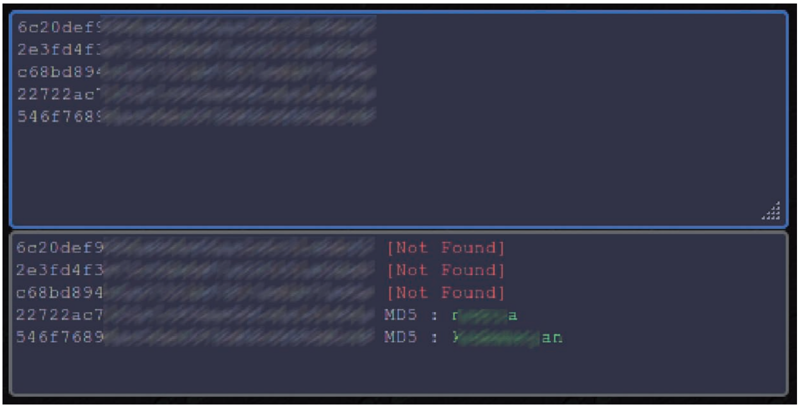
Εικόνα 4
Βρίσκομαι μέσα στην βάση δεδομένων. Ας μελετήσω τους πίνακες...



Εικόνα 5
Να και ο πίνακας με τους χρήστες!

πίνακα χρηστών. Αυτοί οι πίνακες έχουν πάντα στο όνομα τους το string «user", "member" ή κάτι παρόμοιο. Έτσι, με λίγο ψάξιμο είναι εύκολο να τους εντοπίσει κανείς. Όπως αντιλαμβάνεστε, τα δεδομένα ενός τέτοιου πίνακα περιέχουν τα κλειδιά για άλλες πόρτες! Βλέπετε, οι περισσότεροι χρήστες χρησιμοποιούν παρόμοια password --αν όχι τα ίδια ακριβώς-- για τις υπηρεσίες τις οποίες χρησιμοποιούν. Είναι ελάχιστοι εκείνοι που χρησιμοποιούν password managers, με εντελώς διαφορετικά και πολύ δύσκολα passwords για κάθε υπηρεσία. Έτσι, έχοντας το password κάποιου χρήστη, μπορώ κατά πάσα πιθανότητα να συνδεθώ στο email του, στον FTP της εταιρείας και στο Facebook account του! Μην το γελάτε καθόλου. Μιλάμε για μεγάλη πηγή πληροφοριών και, όπως έχετε καταλάβει, στα χέρια αδίστακτων χάκερ, οι πληροφορίες μετατρέπονται εύκολα σε κέρδος. Σκεφτείτε, τώρα, το ενδεχόμενο να μάθουμε το password του administrator. Αν έχει την αφέλεια να χρησιμοποιεί παρόμοια ή κοινά passwords, την έχει πατήσει για τα καλά! Στην εικόνα 5 φαίνεται ο πίνακας με τους χρήστες, τον οποίο εντόπισα στη βάση. Πρώτος και καλύτερος εμφανιζόταν ο administrator! Όμως το password ήταν κρυπτογραφημένο και, με μια γρήγορη ματιά, κατάλαβα ότι πρόκειται για κρυπτογράφηση MD5. Στο δίκτυο υπάρχουν πολλές εφαρμογές που κάνουν MD5 reversing, αξιοποιώντας τεράστιες βάσεις δεδομένων με πιθανά passwords και τα αντίστοιχα MD5 hashes. (Σ.τ.Ε. Βλ. και σχετικό άρθρο στο deltaHacker 008, σελ. 68.) Έτσι, αποφάσισα να δοκιμάσω ν' αντιστρέψω μερικά από τα κρυπτογραφημένα passwords του πίνακα (εικόνα 6).

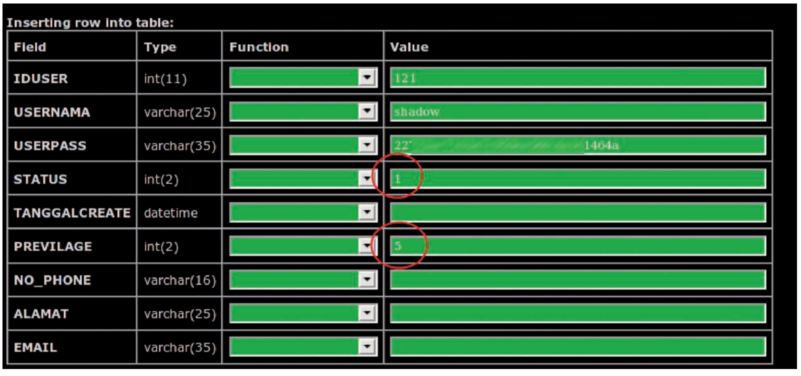
Το ζητούμενο ήταν να σπάσω το password του administrator. Μετά θα μπορούσα να συνδεθώ στην εφαρμογή web ως διαχειριστής κι από εκεί θα αποκτούσα πρόσβαση σε πολλές περισσότερες πληροφορίες! Ωστόσο το password του διαχειριστή ήταν μάλλον δύσκολο και δεν είχα επιτυχία με την αντιστροφή του. Έτσι, στράφηκα σε μια άλλη μέθοδο. Σκέφτη-



Εικόνα 6
Μετά από αρκετό ψάξιμο κατάφερα να "σπάσω" μερικά passwords. Μικρή επιτυχία...

κα να δημιουργήσω ένα πρόσθετο record στον πίνακα με τους χρήστες. Αυτό το record θα αντιστοιχούσε σε έναν χρήστη με τα δικαιώματα του administrator και στο πεδίο του password θα είχε το MD5 hash, ενός γνωστού μου password! Πρόσθεσα μια νέα εγγραφή και φρόντισα τα πεδία "status" και "privilege" να έχουν τις ίδιες τιμές, με τα αντίστοιχα του admin (εικόνα 7). Το username που χρησιμοποιώ συνήθως είναι το shadow. Λίγο κλασικό, θα μου πείτε, αλλά δεν με νοιάζει! Η βάση, πλέον, περιέγραφε έναν χρήστη με το όνομα shadow και τα δικαιώματα του administrator (εικόνα 8). Το επόμενο βήμα ήταν να συνδεθώ στην εφαρμογή web του server, με το νέο μου λογαριασμό. Επομένως, έπρεπε να βρω κάποιο login screen. Εκεί θα έδινα τα credentials του shadow, με την ελπίδα ότι θα συνδεόμουν σαν διαχειριστής. Όπως αντιλαμβάνεστε, το σχετικό login prompt βρέθηκε με ευκολία και, σύντομα, διαπίστωσα ότι το τέχνασμά μου ήταν επιτυχημένο (εικόνες 9 και 10)!

Έχοντας συνδεθεί σαν administrator, επισκέφθηκα το control panel. Μεταξύ άλλων, εκεί βρήκα και τον πίνακα με τους χρήστες, τον οποίο μέχρι πρότινος έβλεπα μόνο μέσα από την ίδια τη βάση (εικόνα 11). Όπως αντιλαμβάνεστε, αυτού του είδους η πρόσβαση άνοιγε πολλούς νέους δρόμους στη δράση της ομάδας WHITE. Χωρίς να το θέλω, επηρεασμένος από όσα κάνω την τελευταία περίοδο, άρχισα να σκέφτομαι πόσα στοιχεία θα μπορούσα να κλέψω, πόσες απάτες θα μπορούσα να στήσω στους υπόλοιπους χρήστες του site και πάει λέγοντας. Πάντως, μη νομίζετε ότι σταμάτησα εκεί. Έχοντας σιγουρέψει την άνετη και χωρίς περιορισμούς πρόσβαση στο site, επέστρεψα στη βάση δεδομένων. Εκεί ανακάλυψα κι


Inserting row into table:

Field	Type	Function	Value
IDUSER	int(11)		121
USERNAMA	varchar(25)		shadow
USERPASS	varchar(35)		22 1404a
STATUS	int(2)		1
TANGGALCREATE	datetime		
PREVILAGE	int(2)		5
NO_PHONE	varchar(16)		
ALAMAT	varchar(25)		
EMAIL	varchar(35)		

Εικόνα 7
Αφού δεν μπορούσα να βρω το password του administrator, σκέφτηκα να φτιάξω έναν κλώνο του -- αλλά με γνωστό password!

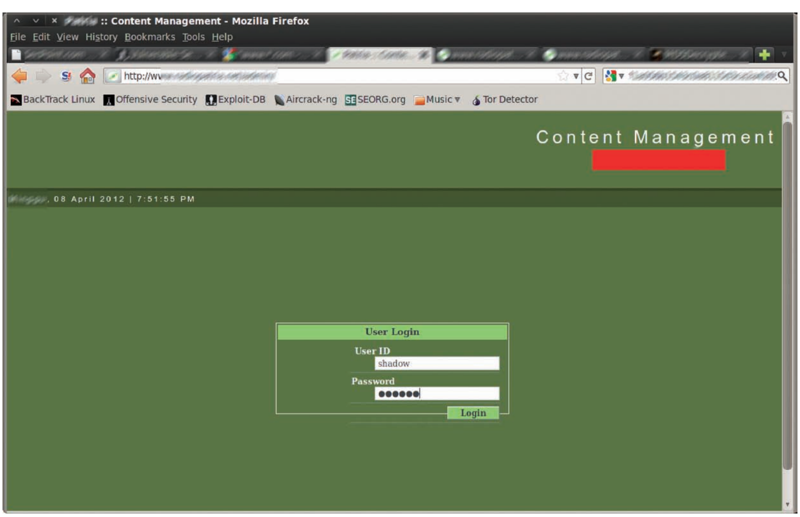
	IDUSER	USERNAMA	USERPASS	STATUS	TANGGALCREATE	PREVILAGE	NO_PHONE	ALAMAT	EMAIL	Action
	1	admin	1f1a[redacted]033	1	[redacted]	5	[redacted]	NULL	NULL	
	12	[redacted]	6c2[redacted]2	NULL	NULL	1	NULL	NULL	NULL	
	3	[redacted]	2e3f[redacted]d3	NULL	NULL	1	[redacted]	NULL	NULL	
	6	[redacted]	c68b[redacted]95e	NULL	NULL	1	NULL	NULL	NULL	
	9	[redacted]	227[redacted]64a	NULL	NULL	1	[redacted]	NULL	NULL	
	10	[redacted]	546f[redacted]48	NULL	NULL	1	[redacted]	NULL	NULL	
	11	[redacted]	6b93[redacted]d84e	NULL	NULL	2	NULL	NULL	NULL	
	121	shadow	227[redacted]64a	1	0000-00-00 00:00:00	5	NULL	NULL	NULL	

Εικόνα 8
Ο νέος χρήστης, με username το shadow, γνωστό password και δικαιώματα πρόσβασης administrator!

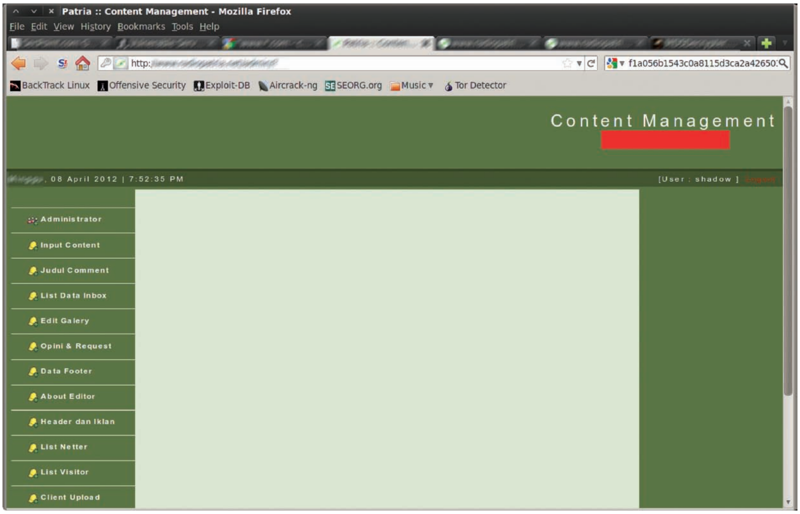
άλλα χρήσιμα δεδομένα. Για παράδειγμα, βρήκα έναν πίνακα με διάφορα προσωπικά στοιχεία των χρηστών (διευθύνσεις email, τηλέφωνα, μηνύματα κ.ά), σε clear text. Όλα αυτά αποτελούσαν λουκουμάκι! Θα έπιαναν καλά λεφτά στη μαύρη αγορά, όπου πολλές "διαφημιστικές" εταιρείες (spammer ολκής) αγοράζουν προσωπικά στοιχεία χρηστών, για να συνθέσουν το προφίλ τους και στη συνέχεια να στείλουν τις κατάλληλες διαφημίσεις. Αυτές οι εταιρείες πληρώνουν καλά και μάλιστα σε bitcoins -- καθόλου άσχημα, ε; (Σ.τ.Ε. Περισσότερα για τα bitcoins διαβάστε στο άρθρο που ξεκινά από τη σελίδα 24 του deltaHacker 008.)

Είμαι ακόμα στην αρχή...

Το ψάξιμο στον συγκεκριμένο server δεν έχει τελειώσει. Σειρά έχουν οι άλλες βάσεις δεδομένων, οι οποίες φιλοξενούνται στον ίδιο database server, ενώ μετά θα πρέπει να προχωρήσω και στις υπόλοιπες υπηρεσίες του μηχανήματος (FTP, email κ.λπ). Μη νομίζετε ότι μετά κι από αυτά θα έχω ξεμπερδέψει. Αντιθέτως, θα έχω πολύ δρόμο μπροστά μου. Συγκεκριμένα, θα έχω τη δύσκολη και χρονοβόρα εργασία της "τυφλής" αναζήτησης στοιχείων στους δίσκους. Αυτή η διαδικασία αφορά σε όλα τα αρχεία που περιέχουν κείμενο (PDF, DOC(X), TXT, XL, PPT κ.λπ.) και θα πρέπει να πραγματοποιηθεί online. Αναρωτιέστε γιατί δεν κατεβάζω τα πάντα στο δικό μου μηχάνημα, ώστε να τα ψάχνω με την ησυχία μου;



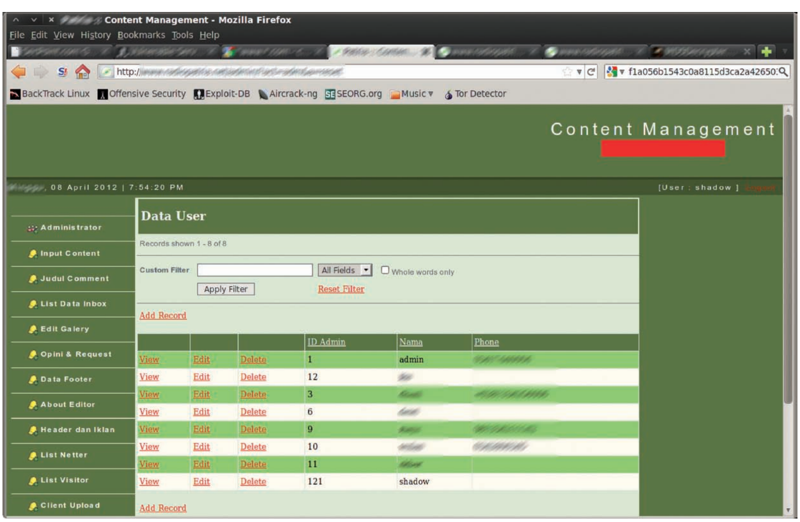
Εικόνα 9
Επιχειρώ να μπω στην εφαρμογή ως administrator...



Εικόνα 10
Howdy, admin! Μόλις συνδεθήκαμε ως άλλος shadow... Ως administrator, δηλαδή!

Αυτό δεν το κάνουμε ποτέ! Αφενός, οι δίσκοι περιέχουν πολλές εκατοντάδες GB και δεν με συμφέρει η μεταφορά τους, αφού θα χάσω πολύ χρόνο. Αφετέρου, ένα τόσο μεγάλο download θα αύξανε *πολύ* τη χρήση του bandwidth κι αυτό θα δημιουργούσε υποψίες στους διαχειριστές. Την ίδια στιγμή υπάρχει και το ενδεχόμενο να εξαντληθεί το bandwidth. Σ' αυτή την περίπτωση η σύνδεση του server διακόπτεται για όλους. Μιλάμε ουσιαστικά για ένα ιδιότυπο DoS attack, το οποίο δεν μας συμφέρει καθόλου.

Τελευταία νέα: Πριν από λίγο έφτασε email για έναν άλλο στόχο. Ενδέχεται να αλλάξουμε αποστολή! Η νέα περίπτωση είναι πολύ πιο προσοδοφόρα για την ομάδα WHITE, αλλά είναι και αρκετά δύσκολη, καθώς έχει πολύ στενά χρονικά περιθώρια... Ελπίζω να μπορέσω να σας γράψω σύντομα!



Εικόνα 11
Ο πίνακας με τους χρήστες, όπως φαίνεται μέσα από το control panel του site.

Skill: Intermediate

Tags: Back up, Incremental backup, Encryption, Encrypted backup, duplicity, duply

how ye bury yer treasure?



by GiannouG

Πώς προστατεύω τα πολύτιμα αρχεία μου; Που τα τοποθετώ και πώς τα μεταφέρω εκεί με ασφάλεια; Κι αν κάποιος ανακαλύψει την κρυψώνα μου; Τι θα συμβεί τότε; Θα παραμείνουν προστατευμένα ή θα μου τα καταστρέψουν/λεηλατήσουν; Προβληματισμοί σαν αυτούς βασανίζουν το μυαλό όλων των φρόνιμων system administrators και users. Κι όχι μόνο: Παρόμοιους είχαν οι πειρατές της Καραϊβικής, όπως κι εκείνοι των θρυλικών παιχνιδιών του Monkey Island. Αλλά ας μην ξεφεύγουμε.

Ένα από τα μεγαλύτερα προβλήματα των admins είναι το backup -- ή τουλάχιστον έτσι θα έπρεπε. Στα τεύχη 005 και 006 του deltaHacker είδαμε πώς μπορούμε ν' αγοράσουμε και να σετάρουμε το δικό μας VPS, από την αρχή μέχρι το τέλος. Δεν είπαμε ωστόσο τίποτα για το backup, ενώ στην πραγματικότητα θα μπορούσαμε να πούμε πολλά.

Μη νομίζετε ότι το backup είναι πάντα μια απλή εργασία. Από τη σωστή λήψη και τήρηση έως και τη διαχείρισή του, απαιτείται αρκετή δουλειά. Ποτέ δεν είναι αρκετό ένα backup και (σχεδόν) πάντα θα υπάρξει κάποιο πρόβλημα, είτε κατά τη διαδικασία επαναφοράς είτε κατά την αποθήκευση στα διάφορα filesystem. Επίσης, το πρόβλημα του backup δεν απασχολεί μόνο τους διαχειριστές συστημάτων. Αποτελεί σημαντικό ζήτημα για *κάθε* χρήστη, τουλάχιστον για όσους δίνουν αξία στα αρχεία και στη δουλειά τους και (σοφά) δεν εμπιστεύονται στο 100% το hardware του υπολογιστή.

Τα πρώτα backup γίνονταν σε κασέτες! Εμείς πάλι θα χρησιμοποιήσουμε λίγο πιο σύγχρονα τεχνικά μέσα, αν και τα tar archives που θα εμπλακούν στην όλη διαδικασία μάς έρχονται απευθείας από το παρελθόν. Δεν είναι εκπληκτικό που ακόμη και σήμερα μάς χρησιμεύουν;



Σ' αυτό το άρθρο δείχνουμε πώς μπορούμε να έχουμε *πάντα* ασφαλή κι αξιόπιστα backup, απλά και γρήγορα. Η λύση που παρουσιάζουμε εφαρμόζεται σε συστήματα Linux, BSD κι άλλα Unixοδειδή, όπως είναι το OS X. Μερικές βασικές χρήσεις που μπορούμε να σκεφτούμε είναι η τήρηση backup του site μας σ' ένα άλλο μηχάνημα (ενδεχομένως VPS), η τήρηση backup του Dropbox folder σ' έναν τοπικό ή απομακρυσμένο fileserver που δεν υποστηρίζει το Dropbox κ.ά.

Ας δούμε όμως τι απαιτείται για ένα καλό backup. Χρειαζόμαστε κατ' αρχάς ένα script, το οποίο θα κρατάει σε tar archives τα κρίσιμα αρχεία. Επίσης, χρήσιμα είναι κι ορισμένα άλλα scripts που θα τρέχουν πριν ή/ και μετά το κύριο backup. Για παράδειγμα, πριν τη λήψη backup από το site μας, ένα script πρέπει να εξάγει τη βάση δεδομένων σε κάποιο αρχείο το οποίο επίσης θα συμμετάσχει στο backup. Να μη ξεχάσουμε και τα timestamps, φυσικά. Κάθε πακέτο αρχείων που καταλήγει να φυλαχτεί στο backup, πρέπει να "γράφει" πάνω του ημερομηνία και ώρα. Έτσι θα είναι εύκολο να το εντοπίσουμε, ενώ θα μπορούμε και να διαγράφουμε τα πολύ παλιά και κατά πάσα πιθανότητα άχρηστα αρχεία. Αν το καλοσκεφτείτε, θα δείτε ότι μιλάμε για μεγάλο παίδεμα. Έτσι είναι, αλλά όχι πια! Όλα αυτά και πολλά περισσότερα αναλαμβάνει να τα κάνει *αυτόματα* το εργαλείο **duplicity**. Παρακάτω θα δούμε την εγκατάσταση και τη ρύθμισή του, ώστε να ξεχάσουμε μια και καλή το ζήτημα του backup!

Πώς το είπες;

Εδώ που τα λέμε, ένας παλιός (και καλός) διαχειριστής συστημάτων έχει σίγουρα τα δικά του εργαλεία backup, τα οποία κατά πάσα πιθανότητα έγραψε μόνος του. Τα εργαλεία αυτά, βέβαια, προσφέρουν πολύ συγκεκριμένες λειτουργίες και είναι απόλυτα προσαρμοσμένα στις ιδιαιτερότητες του εκάστοτε συστήματος. Εμείς δεν θα θέλαμε να γράψουμε κάποιο σχετικό script από την αρχή. Αυτή η δουλειά θα απαιτούσε αρκετό χρόνο και κατά πάσα πιθανότητα το δικό μας script δεν θα προσέφερε τις δυνα-



Δεν πάει να καεί το μηχάνημα; Ποιος νοιάζεται; Με τη βοήθεια των backup που φτιάχνει το duplicity, σε λίγη ώρα θα έχουμε και πάλι τα πολύτιμα αρχεία μας. Σε κάποιο άλλο μηχάνημα, φυσικά :)

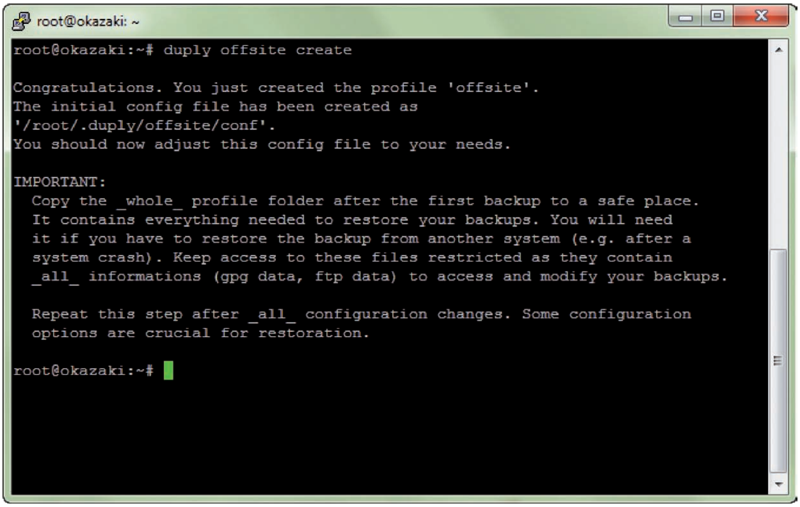
τότητες που παρέχει απλόχερα το duplicity. Αν επιμένετε ότι το δικό μας δημιούργημα θα ήταν καλύτερο, φταίει που δεν ξέρετε τι μπορεί να κάνει το duplicity. Όπως θα έχετε καταλάβει, το εν λόγω εργαλείο δεν κρατάει απλά αντίγραφα. Αντίθετα, διατηρεί πλήρως κρυπτογραφημένα και incremental backups, σε tar archives. Η κρυπτογράφηση επιτυγχάνεται με τη βοήθεια του GnuPG. Με τη συνδρομή του librsync, στα backup αποθηκεύονται *μόνο* τα αρχεία που άλλαξαν ή τροποποιήθηκαν από το προηγούμενο backup (incremental). Νομίζουμε ότι αυτές οι δύο δυνατότητες είναι εξαιρετικά χρήσιμες, δεν συμφωνείτε;

Αν πάντως αδιαφορείτε για την αξία των incremental backups, σκεφτείτε το εξής απλό παράδειγμα. Έστω ότι έχουμε ένα μεγάλο έγγραφο. Ας υποθέσουμε τώρα ότι παίρνουμε ένα backup, τροποποιούμε λίγο το έγγραφο και μετά παίρνουμε νέο backup. Ένα απλό script για backup, το οποίο κάνει απλές αντιγραφές, θα κρατούσε *ολόκληρες* τις εκδοχές του εγγράφου, ενδεχομένως με διαφορετική ημερομηνία στο καθένα. Αργότερα θα μπορούσαμε να τις διακρίνουμε εύκολα και, φυσικά, θα μπορούσαμε να ανασύρουμε (restore) οποιαδήποτε. Όμως στο παράδειγμά μας, το συνολικό μέγεθος του backup ισούται με το άθροισμα των μεγεθών καθεμιάς εκ των εκδοχών του εγγράφου. Για μικρές αλλαγές, είναι φανερό ότι έχουμε μεγάλη σπατάλη αποθηκευτικού χώρου :S Το duplicity, από την άλλη, θα διατηρούσε ένα πλήρες αντίγραφο του αρχικού εγγράφου, ενώ για τη νεότερη εκδοχή θα αποθήκευε *μόνο* τις αλλαγές σε σχέση με την παλιά! Έτσι, θα είχαμε *και* όλες τις ευκολίες (άμεσος εντοπισμός κι επαναφορά οποιουδήποτε στιγμιότυπου) *και* το συνολικό backup θα 'χε μικρότερο μέγεθος! Προφανώς, στην περίπτωση ενός μόνο εγγράφου --κι ανεξαρτήτως του μεγέθους του--, αυτή η διαφορά στον αποθηκευτικό χώρο δεν είναι σημαντική. Στην περίπτωση ωστόσο πολλών εγγράφων, προγραμμάτων, βάσεων δεδομένων κ.λπ., η εξοικονόμηση χώρου είναι σημαντική υπόθεση.

Εγκατάσταση εξοπλισμού

Εντάξει, μας πιάσατε! Ακόμη και το duplicity είναι λίγο πολύπλοκο[1] στη λειτουργία του. Ας μην ξεχνάμε όμως ότι τα backup που δημιουργεί είναι πλήρως κρυπτογραφημένα. Αυτό σημαίνει ότι στο ενδιάμεσο κάθε λειτουργίας του εμπλέκονται κλειδιά, passphrases και άλλα τέτοια ωραία. Εμείς, για να απλοποιήσουμε τη λειτουργία του duplicity, θα επιστρατεύσουμε το duply. Πρόκειται για ένα ανεξάρτητο πρόγραμμα, το οποίο λειτουργεί σαν "χειριστής" (wrapper) του duplicity. Το duply, μεταξύ άλλων, αναλαμβάνει τη διαχείριση των κλειδιών κρυπτογράφησης και μάς γλιτώνει από τον πονοκέφαλο. Επίσης, αυτοματοποιεί διάφορες εργασίες που

1. Και δεν μας αρέσουν τα πολύπλοκα σε απλές δουλειές, έτσι δεν είναι;



Με τη βοήθεια του duply μπορούμε να δημιουργήσουμε πολλές εργασίες backup (backup tasks). Καθεμία από αυτές θα έχει τις δικές της ξεχωριστές ρυθμίσεις και θα μπορούμε να την εκτελούμε ανεξάρτητα από τις άλλες. Η δημιουργία μιας νέας εργασίας γίνεται με την παράμετρο create.

θα θέλαμε να κάνουμε πριν και μετά από τη λήψη ενός backup, για τις οποίες θα έπρεπε να δημιουργούμε πολύπλοκα scripts[2]. Για τις ανάγκες της παρουσίασής μας θα βασιστούμε εξ ολοκλήρου στο Ubuntu. Φυσικά, όσα θα δούμε θα ισχύουν με ελάχιστες διαφορές και σε άλλες διανομές Linux ή BSDs.

Προφανώς, το πρώτο βήμα είναι να ανανεώσουμε την λίστα πακέτων του συστήματος, όπως κάνουμε κάθε φορά πριν από οποιαδήποτε εγκατάσταση πακέτων. Αφού τελειώσει η ανανέωση της λίστας ξεκινάμε με την εγκατάσταση του duplicity και στη συνέχεια του duply:

```
apt-get update
apt-get install duplicity duply
```

Α ναι, δεν μένει να κάνουμε κάτι άλλο :D

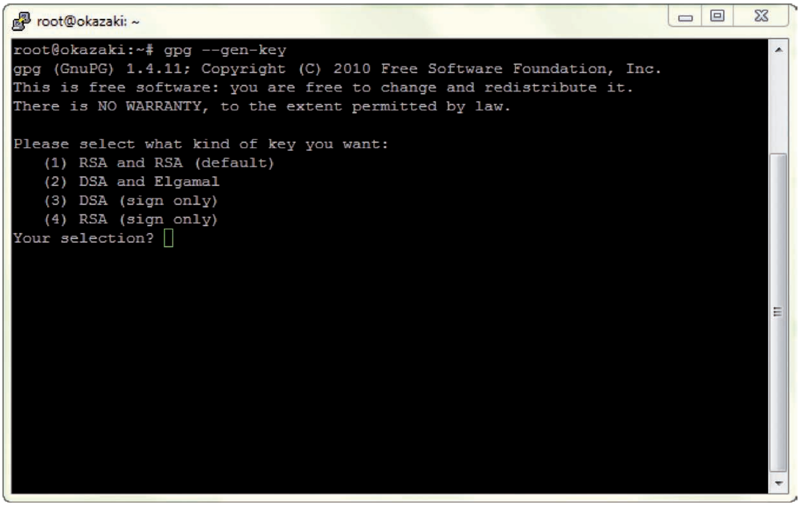
Βασική ρύθμιση

Όπως είπαμε, το duplicity είναι αρκετά πολύπλοκο και γι' αυτό δεν θα το χρησιμοποιήσουμε μόνο του. Για να κάνουμε την ζωή μας ευκολότερη, η ρύθμιση και η χρήση του duplicity θα γίνεται μόνο μέσα από το duply. Για το σετάρισμα μιας εργασίας backup (backup task), χρειάζονται πολλές ρυθμίσεις. Ωστόσο το duply δημιουργεί αυτόματα όλα τα απαραίτητα αρχεία, με τις κατάλληλες πληροφορίες και ρυθμίσεις. Εμείς, αρκεί να δώσουμε κάτι τέτοιο:

```
duply my_treasure create
```

Με αυτόν τον τρόπο παράγονται ορισμένα αρχεία, τα οποία περιγράφουν μια βασική (generic) διαδικασία backup με το duplicity. Επίσης, την πρώτη φορά που θα τρέξουμε το duply, θα δημιουργηθεί και ο κατάλογος ".duply". Αυτός ο κατάλογος βρίσκεται στο home directory του χρήστη, από το λογαριασμό του οποίου εκτελέσαμε το πρόγραμμα. Αν μεταβούμε στον κατάλογο ".duply", μετά την εκτέλεση του παραπάνω, θα βρούμε έναν άλλον με το όνομα my_treasure. Εκεί μέσα υπάρχει το αρχείο ρυθμίσεων για τη νέα μας εργασία backup. Το αρχείο ονομάζεται conf. Ση-

2. Μπορεί να θέλουμε να κάνουμε κάποιες έξτρα δουλίτσες πριν και μετά το κάθε backup. Ουπς, spoiler :S



Το ξεκίνημα της δημιουργίας κλειδιών με το GnuPG. Οι προεπιλεγμένες ρυθμίσεις καλύπτουν απόλυτα τις ανάγκες μας. Αυτό βέβαια δεν σημαίνει ότι δεν πρέπει να πειραματιζόμαστε. Το GnuPG και τις δυνατότητές του μπορείτε να γνωρίσετε από τα σχετικά άρθρα που φιλοξενούμε online: `http://deltahacker.gr/?p=1982` *και* `http://deltahacker.gr/?p=1988`.

μειώστε ότι μπορούμε να δημιουργήσουμε πολλές εργασίες backup (πολλά backup tasks), να επιλέξουμε διαφορετικές ρυθμίσεις για το καθένα και, φυσικά, να εκτελούμε τα tasks σε διαφορετικές χρονικές στιγμές. Όπως αντιλαμβάνεστε, λοιπόν, καλό θα ήταν να προσέχουμε τον κατάλογο ".duply". Χωρίς αυτόν θα είναι δύσκολο να χειριστούμε τα διάφορα backup tasks.

Δημιουργία κλειδιού GnuPG

Κάθε backup είναι κρυπτογραφημένο αλλά *και* υπογεγραμμένο, με ένα κλειδί από το σύστημα κρυπτογράφησης GnuPG (GPG, δείτε και τα `http://deltahacker.gr/?p=1982` και `http://deltahacker.gr/?p=1988`). Αν βέβαια κάποιος θέλει μπορεί να απενεργοποιήσει την κρυπτογράφηση για μια εργασία backup, τροποποιώντας το σχετικό switch στο αρχείο conf της συγκεκριμένης εργασίας. (Σ' εμάς πάντως αρέσει η κρυπτογράφηση!) Πριν προχωρήσουμε, λοιπόν, θα δημιουργήσουμε ένα κλειδί για την κρυπτογράφηση των backup μας.

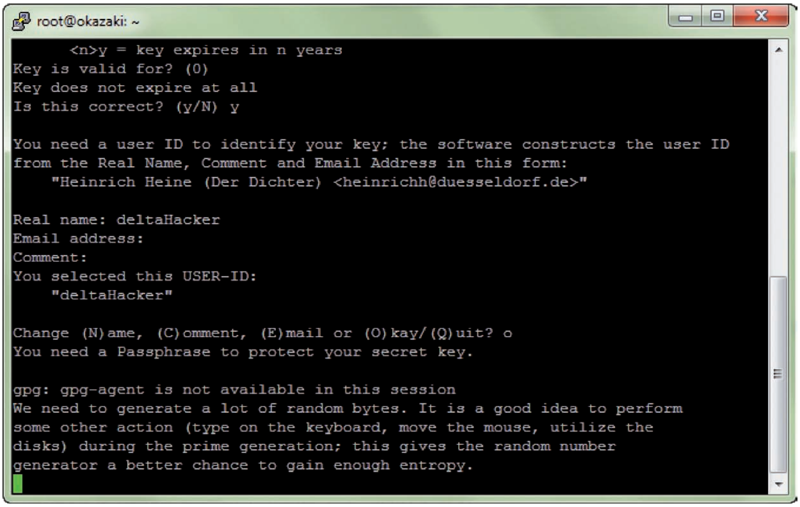
Προφανώς θα χρειαστούμε το πακέτο GnuPG, το οποίο είναι εγκατεστημένο σε (σχεδόν) κάθε διανομή του πλανήτη[3]. Ακόμη κι αν δεν είναι, σίγουρα βρίσκεται διαθέσιμο στα repos για εγκατάσταση. Σε κάθε περίπτωση, εφόσον έχουμε έτοιμα τα σύνεργα ξεκινάμε με τη δημιουργία του πρώτου μας κλειδιού:

```
gpg --gen-key
```

Αρχικά θα εμφανιστούν κάποιες πληροφορίες σχετικά με το πρόγραμμα και θα μας ζητηθεί να επιλέξουμε μια μέθοδο κρυπτογράφησης. Πατάμε Enter για να αποδεχτούμε την προεπιλεγμένη μέθοδο και στη συνέχεια, όταν ζητηθεί το επιθυμητό μέγεθος κλειδιού, αφήνουμε πάλι την προεπιλογή (2048). Ακολούθως, επειδή δεν θέλουμε να λήξει ποτέ το κλειδί μας ορίζουμε ως χρόνο ζωής το 0. Αμέσως μετά πατάμε [Y] για αποδοχή των παραπάνω επιλογών και εισάγουμε ορισμένες πρόσθετες πληροφορίες όπως το όνομα, το username κι ένα email. Πάντως, από αυτές τις

3. Δε ξέρουμε τι γίνεται στο International Space Station, αν και φωτογραφίες δείχνουν ότι τα μηχανήματα τρέχουν Windows :D

Η δημιουργία του κλειδιού απαιτεί εντροπία (σαν να λέμε τυχαιότητα). Στα συστήματα Linux, η εντροπία “δημιουργείται” από τους σκληρούς δίσκους, το ποντίκι και το πληκτρολόγιο...



```
root@okazaki: ~
     <n>y = key expires in n years
Key is valid for? (0)
Key does not expire at all
Is this correct? (y/N) y

You need a user ID to identify your key; the software constructs the user ID
from the Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: deltaHacker
Email address:
Comment:
You selected this USER-ID:
    "deltaHacker"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o
You need a Passphrase to protect your secret key.

gpg: gpg-agent is not available in this session
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
```

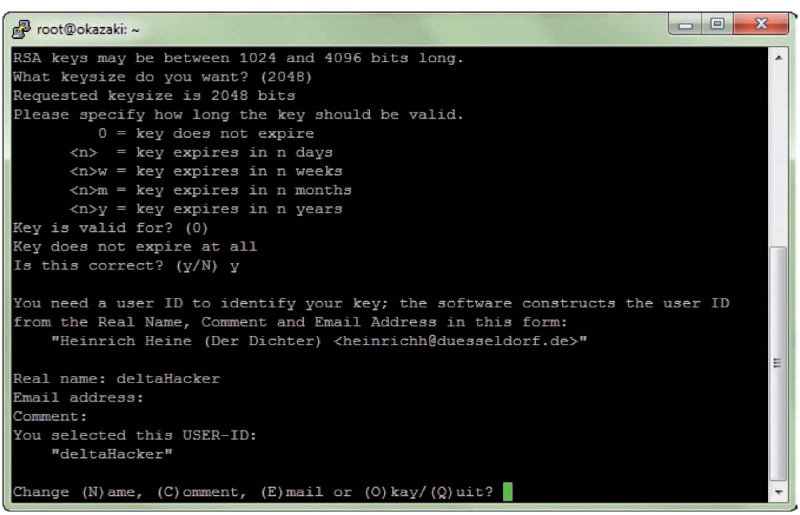
έξτρα πληροφορίες μόνο το πεδίο real name είναι υποχρεωτικό. Τέλος, θα χρειαστεί να δώσουμε ένα passphrase, το οποίο θα πρέπει οπωσδήποτε να θυμόμαστε. Ευτυχώς, δεν χρειάζεται να το απομνημονεύσουμε για όλη μας τη ζωή. Σε λίγο θα το τοποθετήσουμε στο αρχείο ρυθμίσεων του duply και θα ξενοιάσουμε...

Το πρόγραμμα θα προσπαθήσει να μαζέψει λίγη... τύχη[4] κι αφού παρουσιάσει κάποιες πληροφορίες γύρω απ’ το νέο μας κλειδί, θα εμφανίσει και το ίδιο το κλειδί. Η σχετική γραμμή θα είναι κάπως έτσι, αν και το δικό σας κλειδί θα είναι σίγουρα διαφορετικό:

```
gpg: key 23968C75 marked as ultimately trusted
```

Στη δική μας περίπτωση, το κλειδάκι είναι το 23968C75. Τώρα, πρέπει ν’ ανοίξουμε το αρχείο ρυθμίσεων (conf) της εργασίας backup, την οποία

Το μόνο υποχρεωτικό στοιχείο, από αυτά που ζητά το GnuPG, είναι το real name. Φυσικά, εκεί μπορούμε να δώσουμε το αληθινό μας όνομα ή να αφήσουμε τη φαντασία μας ελεύθερη.



```
root@okazaki: ~
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048)
Requested keysize is 2048 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0)
Key does not expire at all
Is this correct? (y/N) y

You need a user ID to identify your key; the software constructs the user ID
from the Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: deltaHacker
Email address:
Comment:
You selected this USER-ID:
    "deltaHacker"

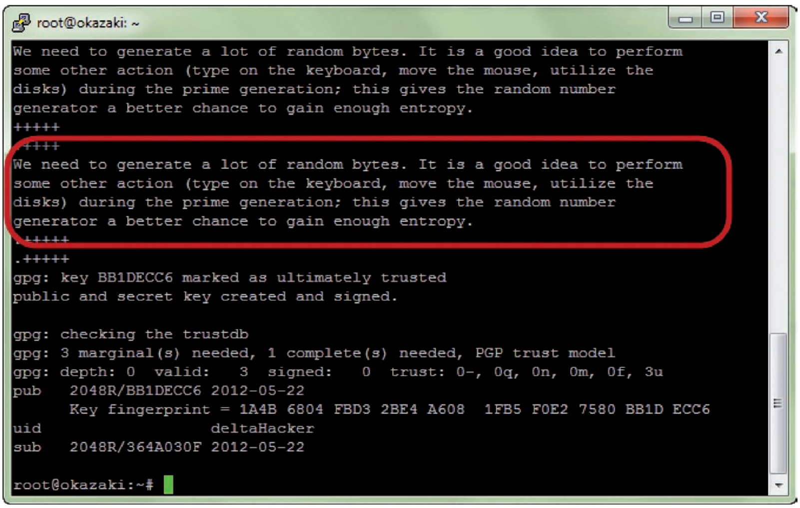
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit?
```

4. Τυχαία δεδομένα, όπως παράσιτα από τους controller των σκληρών δίσκων...

δημιουργήσαμε προηγουμένως (με το όνομα offsite). Για το σκοπό αυτό θα χρησιμοποιήσουμε έναν επεξεργαστή αρχείων απλού κειμένου. Εμείς προτιμάμε το nano στη γραμμή εντολών και το gedit στο περιβάλλον γραφικών, αλλά ο καθένας μπορεί να χρησιμοποιήσει όποιον ξέρει κι αγαπά. Στη γραμμή εντολών, δώσαμε κάτι τέτοιο:

```
nano ~/.duply/my_treasure/conf
```

Σε κάθε αρχείο ρυθμίσεων του duply, οι πρώτες γραμμές περιέχουν πληροφορίες για τα κλειδιά κρυπτογράφησης της εκάστοτε εργασίας backup. Εμείς θα πρέπει να εισαγάγουμε το κλειδί μας στο πεδίο GPG_KEY και το passphrase στο πεδίο GPG_PW. Αυτό ήταν! Τώρα μπορούμε να προχωρήσουμε στην ουσιαστική ρύθμιση του backup task.



Όταν το GnuPG πρόγραμμα "συλλέξει" την εντροπία που χρειάζεται, θα δημιουργήσει και θα εμφανίσει το νέο μας κλειδί.

Απομακρυσμένη αποθήκευση

Αρκετές εταιρίες hosting προσφέρουν ως *δώρο* την πρόσβαση σε κάποιον FTP server, αποκλειστικά και μόνο για τη διατήρηση backup. Βέβαια, εδώ που τα λέμε, δεν είναι και δύσκολο να δημιουργήσει κανείς τον δικό του FTP server, σε κάποιο VPS. Πάντως, εφόσον τα backup μας είναι κρυπτογραφημένα, δεν μας πειράζει καθόλου ν' αποθηκεύονται σε δημόσιους FTP severs.

Η επιλογή για το που θα αποθηκεύονται τα backup, γίνεται από το πεδίο TARGET του αρχείου ρυθμίσεων. Παρακάτω φαίνονται δύο διαφορετικοί τρόποι δήλωσης του πεδίου, για αποθήκευση των backup σε κάποιον FTP sever. Η μόνη διαφορά μεταξύ των δύο δηλώσεων αφορά στη χρήση credentials, για την είσοδο στο server.

```
TARGET='ftp://ftp.giannoug.gr/backups'
TARGET='ftp://giannoug:password@ftp.giannoug.gr/backups'
```

Με την ίδια ευκολία, μπορούμε να ορίσουμε και κάποιον SSH server. Η λογική είναι ακριβώς η ίδια. Το μόνο που αλλάζει είναι το πρωτόκολλο επικοινωνίας:

```
TARGET='ssh://giannoug:pass@giannoug.gr/backups'
```

Αποθήκευση στο mail!

Μην πάει το μυαλό σας σε κάνα «βρώμικο» τερτίπι. Στο μυαλό μας έχουμε κάτι απλό και απόλυτα λειτουργικό. Όταν πρωτοεμφανίστηκε το Google Mail, αναρωτιόμασταν τι στο καλό χρειάζονται τόσα πολλά gigabyte και μάλιστα δωρεάν. Εδώ που τα λέμε, ο μέσος χρήστης δεν καταναλώνει ούτε καν το 10% του διαθέσιμου χώρου στο Gmail, αφού τα ίδια τα email καταλαμβάνουν ελάχιστο χώρο. Το duplicity, πάντως, επιτρέπει να αποθηκεύουμε αρχεία ακόμα και σε καταλόγους IMAP! (Intetnet Message Access Protocol, http://en.wikipedia.org/wiki/Internet_Message_Access_Protocol) Καλό, έτσι; Αναφερθήκαμε στο Gmail, μόνο και μόνο γιατί είναι η διασημότερη υπηρεσία free mail και γιατί προσφέρει άφθονο αποθηκευτικό χώρο, καθώς και πρόσβαση IMAP. Στην πραγματικότητα, υπάρχουν αρκετές ακόμα υπηρεσίες που προσφέρουν αυτές τις ευκολίες. Ένα παράδειγμα ρύθμισης του duply (κι επομένως του duplicity) για χρήση IMAP, φαίνεται παρακάτω:

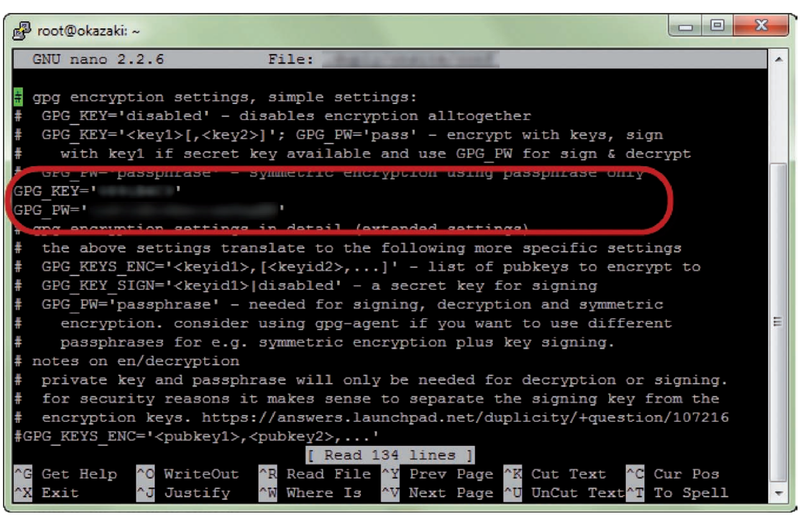
```
TARGET='imaps://username:password@imap.gmail.com/'
```

Έτσι, τα αρχεία θα ανεβαίνουν στον IMAP server της Google και θα μπορούμε να τα βλέπουμε ακόμα και μέσα από το web interface του Gmail, καθώς κι από οποιονδήποτε IMAP client. Εδώ βέβαια οφείλουμε να σημειώσουμε ότι κάτι τέτοιο ενδέχεται να μην αρέσει στη Google. Επομένως, δεν αποκλείεται να συνδεθούμε κάποια μέρα και τα backup να έχουν εξαφανιστεί :S

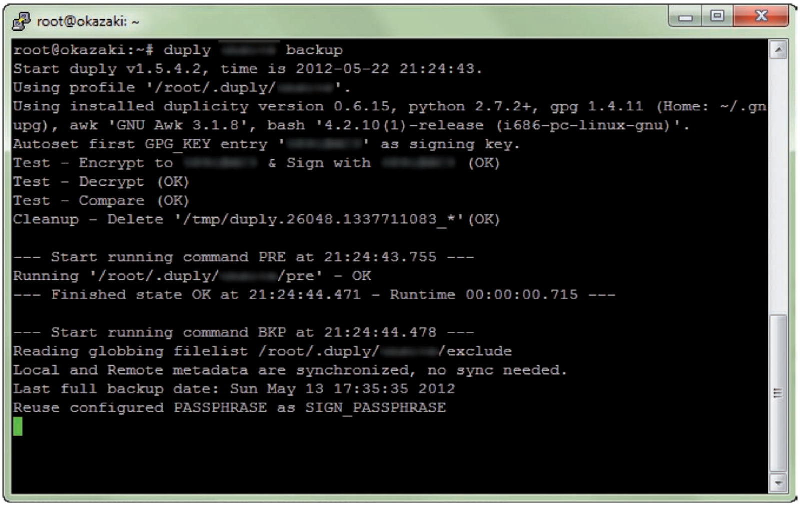
Τοπική αποθήκευση

Ναι, το ξέρουμε! Η ιδέα της τοπικής αποθήκευσης των backup δεν μοιάζει και πολύ καλή. Ωστόσο, τα backup δεν μας προφυλάσσουν αποκλειστικά από την αστοχία του hardware. Είναι πιο πιθανό να καταστραφεί ένα αρχείο με δική μας υπαιτιότητα, παρά γιατί κάποιος δίσκος αποφάσισε να παραδώσει πνεύμα στις 5 το πρωί, ξημερώματα Σαββάτου. Ας υποθέσουμε, λοιπόν, ότι θέλουμε τα πακέτα του backup να αποθηκεύονται στον κατάλογο backup (πολύ πρωτότυπο, έτσι;). Το μόνο που πρέπει να κάνουμε, είναι να ορίσουμε το πεδίο TARGET κάπως έτσι:

```
TARGET='file:///home/giannoug/backup'
```



Το κλειδί του GnuPG και το αντίστοιχο passphrase, πρέπει να τοποθετηθούν στα κατάλληλα πεδία του αρχείου ρυθμίσεων του backup task που φτιάξαμε με το duply.



Το duply εν ώρα εργασίας. Τα πολύτιμα αρχεία μας πρόκειται να αντιγραφούν σε μια μακρινή τοποθεσία. Κρυπτογραφημένα, φυσικά!

Για ποιον όλα αυτά;

Για το τέλος αφήσαμε την απλούστερη αλλά και πιο χρήσιμη επιλογή: Τον προσδιορισμό του καταλόγου, τον οποίο θέλουμε να προστατεύσουμε. Επιπλέον, θα δούμε ορισμένες πρόσθετες δυνατότητες που προσφέρει το duply. Λίγο χαμηλότερα από το πεδίο TARGET και πάντα εντός του αρχείου conf, θα βρούμε το πεδίο SOURCE. Σε αυτό θα καθορίσουμε τον κατάλογο που θέλουμε να προφυλάσσεται (δηλαδή τον κατάλογο τον οποίο θα παίρνουμε backup). Ας υποθέσουμε, λοιπόν, ότι ο πολυτιμότερος κατάλογος του συστήματός μας είναι το document root του site μας με όνομα my_super_site (Σ.τ.Ε. Ναι, η πρωτοτυπία σταματημό δεν έχει :) Σε αυτή την περίπτωση --και με την προϋπόθεση ότι ο web server τρέχει στο μηχάνημα στο οποίο ρυθμίζουμε το duply-- αρκεί να δώσουμε κάτι τέτοιο:

```
SOURCE='/var/www/my_super_site'
```

Αυτό ήταν! Όλα τα αρχεία του φακέλου που ορίσαμε, θα αποθηκεύονται αυτόματα. Εδώ όμως, προκύπτει ένα ζήτημα: Τι μπορούμε να κάνουμε, αν θέλουμε να αποθηκεύονται κι άλλα αρχεία, τα οποία βρίσκονται εκτός του συγκεκριμένου καταλόγου; Παράδειγμα: Πώς θα κρατήσουμε αντίγραφο *και* από τη βάση δεδομένων του site μας;

Το duply προβλέπει την αυτόματη εκτέλεση δύο συγκεκριμένων script, πριν και μετά την διαδικασία ενός backup. Συγκεκριμένα, πριν από τη διαδικασία του backup αναζητά ένα script με το όνομα pre κι εφόσον το εντοπίσει, το εκτελεί. Αντίστοιχα, μετά από τη διαδικασία του backup, αναζητά ένα script με το όνομα post κι εφόσον το βρει, επίσης το εκτελεί. Για την κατασκευή αυτών των δύο script είμαστε υπεύθυνοι εμείς και θα πρέπει να τα τοποθετήσουμε στον κατάλογο του backup task, μαζί με το αντίστοιχο conf. Στην περίπτωσή μας, θα πρέπει να τα τοποθετήσουμε στον κατάλογο "~/.duply/treasure/". Για την αποθήκευση της βάσης, λοιπόν, θα αρκούσε να γράψουμε ένα κατάλληλο script, με το όνομα pre. Αναρωτιέστε τι εννοούμε με το κατάλληλο; Φανταστείτε ένα script στο οποίο χρησιμοποιείται το mysqldump για την εξαγωγή ενός συμπιεσμένου archive της MySQL, εντός του document root του site μας. Παράδειγμα:

```
/usr/bin/mysqldump --add-drop-table -h localhost \
-u όνομα_χρήστη_βάσης -pκωδικός_χρήστη_βάσης \
όνομα_βάσης | bzip2 -9 -c > \
/var/www/my_super_site/όνομα_βάσης.sql.bz2
```

Βέβαια, μαζί μ' αυτό το script θα ήταν καλή ιδέα να φτιάξουμε κι ένα ακόμα, με το όνομα post. Αυτό το script, το οποίο θα εκτελείται μετά τη λήψη του backup, θα διαγράφει τη βάση. Συνεχίζοντας το προηγούμενο παράδειγμα, το περιεχόμενο του post θα μπορούσε να είναι το ακόλουθο:

```
rm -fr /var/www/my_super_site/όνομα_βάσης.sql.bz2
```

Λειτουργία και αυτοματοποίηση

Μπορούμε να κρατάμε backup των αρχείων μας οποτεδήποτε. Αρκεί να τρέχουμε το duply με πρώτη παράμετρο το όνομα της επιθυμητής εργασίας backup και δεύτερη παράμετρο τη λέξη backup. Παράδειγμα:

```
duply treasure backup
```

Φυσικά, τις περισσότερες φορές, θα θέλουμε να λαμβάνονται αυτόματα τα backup. Για το σκοπό αυτό θα πρέπει να στραφούμε στο crontab. Αρχικά, δεν έχουμε παρά να δώσουμε κάτι τέτοιο:

```
sudo crontab -e
```

Θ' ανοίξει έτσι ο default text editor και θα δούμε τις ήδη προγραμματισμένες εργασίες του συστήματός μας. Αν επιθυμούμε την αυτόματη λήψη backup καθημερινά και συγκεκριμένα τα μεσάνυχτα, στο τέλος αυτού του αρχείου θα πρέπει να προσθέσουμε την ακόλουθη γραμμή:

```
@midnight /usr/bin/duply offsite backup
```

Αμέσως μετά μπορούμε να αποθηκεύσουμε τις αλλαγές μας και να βγούμε από τον text editor. Το crontab θα είναι πλέον ρυθμισμένο, ώστε να εκτελεί αυτόματα τη διαδικασία του backup τα μεσάνυχτα. Στην περίπτωση που ακούτε πρώτη φορά για το crontab ή για το @midnight, θα σας προτείναμε να διαβάσετε τα man pages των cron και anacron (man cron και man anacron αντίστοιχα).

Ανάκτηση

Με τη βοήθεια του duply μπορούμε πολύ εύκολα ν' ανακτήσουμε ένα ολόκληρο backup ή μεμονωμένα αρχεία. Για την πρώτη περίπτωση το duply διαθέτει την επιλογή restore, ενώ για τη δεύτερη προσφέρεται η επιλογή fetch. Όλα αυτά θα φανούν καλύτερα με δύο παραδείγματα. Ας υποθέσουμε αρχικά ότι θέλουμε ν' ανακτήσουμε ολόκληρο το backup και να τοποθετηθεί στον κατάλογο ~/temp. Γι' αυτή την εργασία αρκεί να δώσουμε κάτι τέτοιο:

```
duply my-treasure restore ~/temp
```

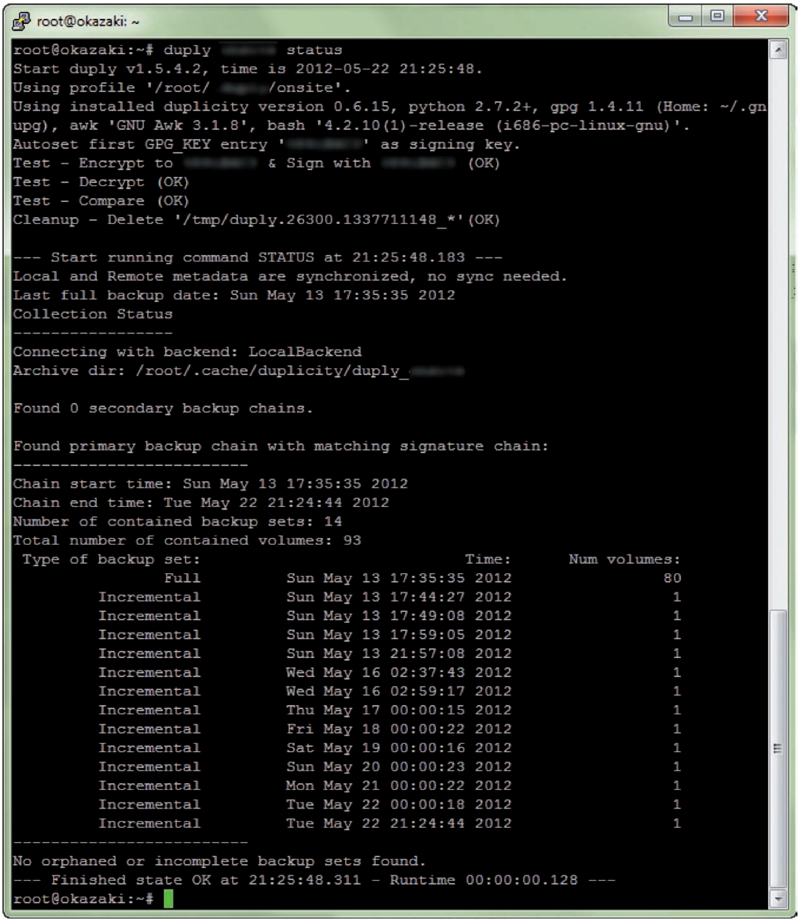
Τώρα, ας υποθέσουμε ότι θέλουμε να ανακτήσουμε ένα μεμονωμένο αρχείο. Έστω ότι μας ενδιαφέρει το αρχείο form.html, το οποίο βρίσκεται στη θέση /var/www/my_super_site/html. Ας υποθέσουμε, επίσης, ότι το αντίγραφο που θ' ανακτηθεί, θέλουμε τοποθετηθεί στον κατάλογο ~/work. Για να λάβουμε την πιο πρόσφατα αποθηκευμένη έκδοση του αρχείου, αρκεί να δώσουμε αυτό:

```
duply my-treasure fetch /var/www/my_super_site/html/form.html \
~/work/form.html
```

Φυσικά, οι ευκολίες δεν σταματούν εδώ. Εφόσον τα backup του duplicity είναι incremental, μπορούμε εύκολα να ζητήσουμε παλιότερες εκδοχές των αρχείων μας. Έτσι, στην περίπτωση που θέλαμε πάλι το ίδιο αρχείο, αλλά στη μορφή που είχε τέσσερεις μέρες νωρίτερα, θα δίναμε αυτό:

```
duply my-treasure fetch /var/www/my_super_site/html/form.html \
~/work/form.html 4D
```

Όπως αντιλαμβάνεστε, οι δυνατότητες των duply και duplicity είναι αναρίθμητες. Περισσότερες επιλογές, ρυθμίσεις και κολπάκια, θα βρείτε στις αντίστοιχες man pages. Επίσης, μην ξεχάσετε να διαβάσετε τα σχόλια του αρχείου conf, στον κατάλογο του backup task που θα φτιάξετε.



```
root@okazaki: ~
root@okazaki:~# duply [redacted] status
Start duply v1.5.4.2, time is 2012-05-22 21:25:48.
Using profile '/root/[redacted]/onsite'.
Using installed duplicity version 0.6.15, python 2.7.2+, gpg 1.4.11 (Home: ~/.gn
upg), awk 'GNU Awk 3.1.8', bash '4.2.10(1)-release (i686-pc-linux-gnu)'.
Autoset first GPG_KEY entry '[redacted]' as signing key.
Test - Encrypt to [redacted] & Sign with [redacted] (OK)
Test - Decrypt (OK)
Test - Compare (OK)
Cleanup - Delete '/tmp/duply.26300.1337711148_*'(OK)

--- Start running command STATUS at 21:25:48.183 ---
Local and Remote metadata are synchronized, no sync needed.
Last full backup date: Sun May 13 17:35:35 2012
Collection Status
-----------------
Connecting with backend: LocalBackend
Archive dir: /root/.cache/duplicity/duply_[redacted]

Found 0 secondary backup chains.

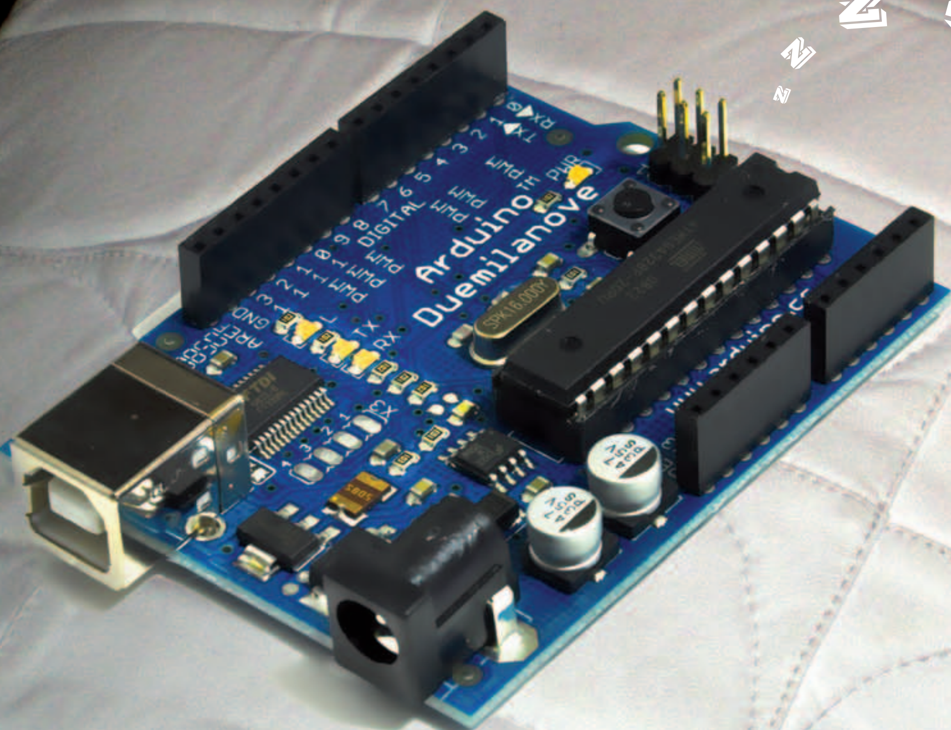
Found primary backup chain with matching signature chain:
-------------------------
Chain start time: Sun May 13 17:35:35 2012
Chain end time: Tue May 22 21:24:44 2012
Number of contained backup sets: 14
Total number of contained volumes: 93
 Type of backup set:                            Time:      Num volumes:
                Full         Sun May 13 17:35:35 2012                80
         Incremental         Sun May 13 17:44:27 2012                 1
         Incremental         Sun May 13 17:49:08 2012                 1
         Incremental         Sun May 13 17:59:05 2012                 1
         Incremental         Sun May 13 21:57:08 2012                 1
         Incremental         Wed May 16 02:37:43 2012                 1
         Incremental         Wed May 16 02:59:17 2012                 1
         Incremental         Thu May 17 00:00:15 2012                 1
         Incremental         Fri May 18 00:00:22 2012                 1
         Incremental         Sat May 19 00:00:16 2012                 1
         Incremental         Sun May 20 00:00:23 2012                 1
         Incremental         Mon May 21 00:00:22 2012                 1
         Incremental         Tue May 22 00:00:18 2012                 1
         Incremental         Tue May 22 21:24:44 2012                 1
-------------------------
No orphaned or incomplete backup sets found.
--- Finished state OK at 21:25:48.311 - Runtime 00:00:00.128 ---
root@okazaki:~#
```

Με την εντολή status το duply μάς δείχνει την κατάσταση του backup. Η συγκεκριμένη εργασία τρέχει εδώ και μερικές ημέρες. Έτσι, έχουν ληφθεί αρκετά incremental backups και στην εικόνα φαίνεται μία λίστα με όλα τα αποθηκευμένα set...

Skill: Intermediate

Tags: Binary Code Modulation, BCM, sleep modes, interrupts, timers, fading, morphing

Arduino
the way of the hacker!!
(Μέρος 6/k*)



by Spir@lEvolution

Καλό και χρυσό το Pulse Width Modulation, αλλά όταν το υλοποιούμε σε software παρενοχλεί τον επεξεργαστή ασταμάτητα και σε πολύ μικρά χρονικά διαστήματα. Ούτε έναν υπνάκο της προκοπής δεν τον αφήνει να πάρει! Ευτυχώς, στις περισσότερες περιπτώσεις μπορούμε να αντικαταστήσουμε το PWM με μια άλλη τεχνική, η οποία απασχολεί τον επεξεργαστή πολύ λιγότερο και είναι εξαιρετικά αποδοτική: Το Binary Code Modulation. Με την ευκαιρία θα σας πούμε και για εκείνον τον υπνάκο, τον οποίο θα ήθελε να απολαμβάνει κάθε τόσο ο επεξεργαστής...

*το k δεν έχει φανταστικό μέρος

Στο προηγούμενο άρθρο της σειράς μας πάνω στο Arduino, που δημοσιεύσαμε στο deltaHacker 007, καταπιαστήκαμε με το θέμα του Pulse Width Modulation. Ξεκινήσαμε μελετώντας τα χαρακτηριστικά ενός σήματος PWM και συσχετίσαμε το Duty Cycle με τη μέση τιμής της τάσης. Έτσι, κατανοήσαμε πώς είναι δυνατό να σχηματίζουμε οποιαδήποτε τάση στην περιοχή από 0 έως 5Volt, χρησιμοποιώντας ένα σήμα το οποίο κάθε στιγμή έχει είτε τη μία είτε την άλλη τιμή. Φυσικά, όταν ξεμπερδέψαμε με τη θεωρία προχωρήσαμε στην υλοποίηση. Εξετάζοντας το σύστημα PWM που ενσωματώνει ο επεξεργαστής του Arduino, καταλήξαμε σε μια ιδιαίτερα αποδοτική υλοποίηση του PWM σε software. Όλα αυτά δεν τα κάναμε στο πλαίσιο κάποιας ανούσιας άσκησης. Το hardware του Arduino προσφέρει μόλις έξι κανάλια PWM, ενώ σε αρκετά project ενδέχεται να χρειαστούμε περισσότερα. Στο τέλος, λοιπόν, κατασκευάσαμε ένα σύστημα δέκα καναλιών PWM. Μάλιστα το εν λόγω σύστημα μπορούσε να επεκταθεί πολύ εύκολα, ώστε να προσφέρει ακόμη περισσότερα κανάλια.

Too good to be... good

Το Pulse Width Modulation έχει πολλές εφαρμογές: Από τη ρύθμιση της φωτεινότητας ενός LED και της ταχύτητας ενός κινητήρα, μέχρι τον έλεγχο ενός μετατροπέα τάσης (buck ή boost converter) και τη σύνθεση ήχου! Εξάλλου, η υλοποίησή του σε software αποτελεί αρκετά απλή υπόθεση. Ειδικά τότε, όμως, συνοδεύεται από ένα βαρύ αντίτιμο: Την αυξημένη κατανάλωση επεξεργαστικής ισχύος και κατ' επέκταση την αυξημένη κατανάλωση ενέργειας. Για του λόγου το αληθές, θα υπολογίσουμε τις επιδόσεις μιας υλοποίησης του PWM στο software, σαν αυτή που συναντήσαμε στο τέλος του άρθρου στο deltaHacker 007. Σ' εκείνο το πρόγραμμα ο μηχανισμός του PWM βρισκόταν στη ρουτίνα εξυπηρέτησης ενός interrupt. Αυτή η ρουτίνα επαναλαμβανόταν αυτούσια σε τακτά χρονικά διαστήματα, διακόπτοντας κάθε άλλη εργασία του συστήματος. Επομένως, για να βρούμε την επίδραση του PWM στις επιδόσεις του προγράμματος αρκεί να εξετάσουμε τη συγκεκριμένη ρουτίνα. Το σχετικό interrupt ήταν κατάλληλα ρυθμισμένο, ώστε να προκύπτει κάθε στιγμή κατά την οποία έπρεπε να ελεγχθεί (και ενδεχομένως να μεταβληθεί) η στάθμη κάθε καναλιού. Ας υποθέσουμε, λοιπόν, ότι θέλουμε να φτιάξουμε ένα σύστημα PWM με συχνότητα 400Hz[1] και με 256 στάθμες ρύθμισης. Σε αυτή την περίπτωση, το interrupt που αναφέραμε προηγουμένως θα πρέπει να εκτελείται 256 φορές, σε κάθε περίοδο του PWM. Αυτό σημαίνει ότι θα πρέπει να εκτελείται με συχνότητα 256 * 400Hz = 102,4KHz! Με άλλα λόγια, το εν λόγω interrupt θα πρέπει να εκτελείται ανά σχεδόν 9,7μsec. Σ' αυτό το χρονικό διάστημα ο επεξεργαστής του Arduino προλαβαίνει να εκτελέσει περί τις 155 εντολές[2] της γλώσσας μηχανής. Αν υποθέσουμε ότι η ρουτίνα του PWM ολοκληρώνεται σε 40 εντολές της γλώσσας μηχανής και αν συνυπολογίσουμε τις καθυστερήσεις για την διακοπή του προγράμματος και την εκτέλεση του interrupt service routine, το σύστημα PWM δεσμεύει σχεδόν το 30% του χρόνου του επεξεργαστή! Αυτό το ποσοστό είναι απαράδεκτα υψηλό και καθιστά το software PWM "ασύμβατο" με όλες τις εφαρμογές, οι οποίες απαιτούν αυξημένη επεξεργαστική ισχύ.

Ας δούμε τι συμβαίνει και με την κατανάλωση ενέργειας. Ο επεξεργαστής του Arduino διαθέτει ορισμένες καταστάσεις λειτουργίας, κατά τις οποί-

1. Αυτή η χαμηλή συχνότητα για το PWM είναι αποδεκτή για λίγες και απλές εφαρμογές. Σε πιο προχωρημένες εργασίες απαιτούνται *πολύ* μεγαλύτερες συχνότητες.

2. Αυτό συμβαίνει γιατί ο επεξεργαστής του Arduino τρέχει στα 16MHz και οι περισσότερες εντολές του εκτελούνται σε έναν κύκλο του ρολογιού (62,5nsec ανά εντολή).

ες απενεργοποιούνται διάφορα κυκλώματά του. Αυτές οι καταστάσεις ονομάζονται sleep modes και η ύπαρξή τους αποσκοπεί στην εξοικονόμηση ενέργειας. Όπως υποψιάζεστε, τα sleep modes διαφοροποιούνται ως προς την απόδοσή τους. Κάποια κοιμίζουν ελαφρά τον επεξεργαστή και δεν εξοικονομούν αρκετή ενέργεια, ενώ άλλα τον κοιμίζουν βαθειά και περιορίζουν αρκετά την κατανάλωση. Η επαναφορά του επεξεργαστή από κάποιο "βαθύ" sleep mode δεν πραγματοποιείται αμέσως. Συνήθως, μεσολαβεί μια καθυστέρηση της τάξης των μsec. Αυτό το χρονικό διάστημα, όπως και η ίδια η διαδικασία της ενεργοποίησης, απαιτεί ενέργεια. Κατά συνέπεια, ένας συνεχόμενος ύπνος είναι πιο αποδοτικός σε σχέση με έναν διακεκομμένο, της ίδιας συνολικής διάρκειας. Έτσι, ένα interrupt το οποίο εκτελείται με μεγάλη συχνότητα και ξυπνάει τον επεξεργαστή κάθε λίγο και λιγάκι επιδρά αρνητικά στην κατανάλωση του συστήματος. Το software PWM, λοιπόν, αποδεικνύεται και ενεργειακά σπάταλο...

Ύπνε που παίρνεις τα παιδιά...

...πάρε και τον επεξεργαστή μου! Ο επεξεργαστής του Arduino υποστηρίζει έξι διαφορετικά "επίπεδα" ύπνου (sleep modes). Σε καθένα από αυτά απενεργοποιείται ένα διαφορετικό σύνολο υποσυστημάτων και επιτυγχάνεται μεγαλύτερη ή μικρότερη εξοικονόμηση ενέργειας. Αν θέλετε να μάθετε λεπτομέρειες για το ποιο σύστημα απενεργοποιείται σε κάθε sleep mode, μπορείτε να συμβουλευτείτε το datasheet του επεξεργαστή (`www.atmel.com/Images/doc8271.pdf`) και συγκεκριμένα τις σελίδες 40 έως 42. Παρακάτω φαίνονται τα έξι sleep modes, ταξινομημένα με αύξουσα σειρά ως προς την εξοικονόμηση ενέργειας που πετυχαίνουν:

- Idle (SLEEP_MODE_IDLE)
- ADC Noise Reduction (SLEEP_MODE_ADC)
- Power-save (SLEEP_MODE_PWR_SAVE)
- Standby (SLEEP_MODE_STANDBY)
- Extended Standby (SLEEP_MODE_EXT_STANDBY)
- Power-down (SLEEP_MODE_PWR_DOWN)

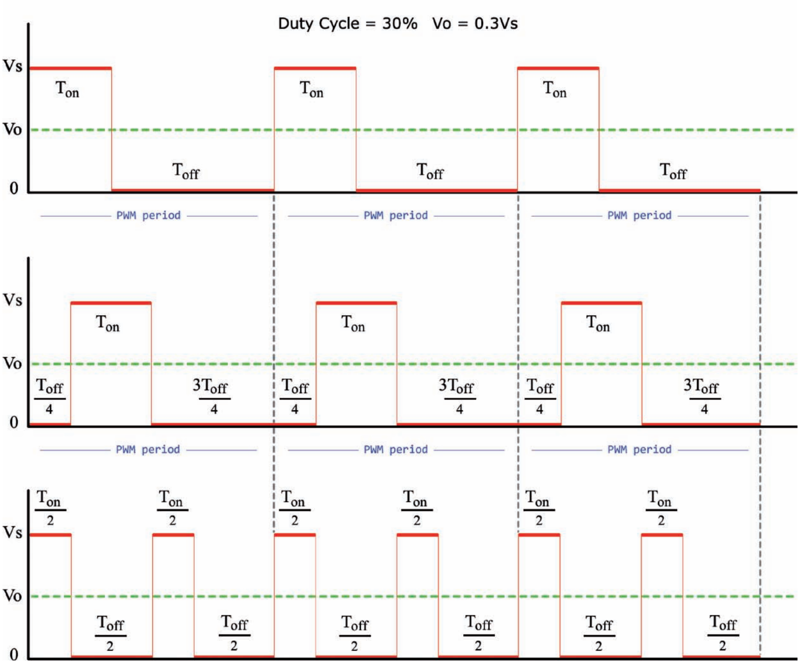
Για τη χρήση ενός sleep mode σε κάποιο πρόγραμμα του Arduino, πρέπει αρχικά να ενσωματώσουμε τη σχετική βιβλιοθήκη. Αυτό επιτυγχάνεται περιλαμβάνοντας στο πρόγραμμα το αντίστοιχο header file. Στη συνέχεια πρέπει να καθορίσουμε το επίπεδο "ύπνωσης", στο οποίο θα περιέρχεται ο επεξεργαστής. Αυτό επιτυγχάνεται με τη συνάρτηση set_sleep_mode. Σε αυτή δίνουμε το όνομα του επιθυμητού sleep mode, όπως εμφανίζεται στον παραπάνω πίνακα και *μέσα στις παρενθέσεις*. Ακολούθως πρέπει να εκτελέσουμε τη συνάρτηση sleep_enable, ώστε να ενεργοποιηθεί η δυνατότητα "ύπνωσης" του επεξεργαστή.

```
#include <avr/sleep.h>
set_sleep_mode(SLEEP_MODE_EXT_STANDBY);
sleep_enable();
```

Πλέον, οποτεδήποτε θέλουμε να κοιμηθεί ο επεξεργαστής αρκεί να καλούμε τη συνάρτηση sleep_mode.

Binary Code Modulation

Είναι φανερό ότι τα μειονεκτήματα του software PWM πηγάζουν από την εξαντλητικά συχνή εκτέλεση ενός interrupt service routine. Αναφερόμαστε στη ρουτίνα η οποία ελέγχει τη στάθμη κάθε καναλιού και πραγματοποιεί τις απαιτούμενες μεταβολές. Η συχνότητα εκτέλεσης αυτού του κώδικα καθορίζεται από την αναλυτικότητα[3] και τη συχνότητα του PWM που υλοποιούμε. Επομένως, αν επιθυμούμε την ελάχιστη δυνατή επιβάρυνση του επεξεργαστή και την ελάχιστη ενεργειακή κατανάλωση, πρέπει να υλοποιήσουμε ένα software PWM με πολύ φτωχά χαρακτηριστικά ή, πολύ απλά, να το ξεχάσουμε! Κάπως έτσι έρχεται στο προσκήνιο μια τεχνική με το όνομα Binary Code Modulation (BCM), την οποία ενδέχεται να συναντήσετε και ως Bit Angle Modulation (BAM). Με αυτή την τεχνική μπορούμε να ρυθμίζουμε την τάση επάνω σε ψηφιακούς ακροδέκτες, όπως και με το PWM. Ωστόσο με το BCM μπορούμε να έχουμε μεγάλη αναλυτικότητα στη ρύθμιση, μεγάλη συχνότητα λειτουργίας[4] και, την ίδια στιγμή, χαμηλή επιβάρυνση στον επεξεργαστή κι ελάχιστη κατανάλωση! Για να αποκτήσετε μια αίσθηση των προσόντων του BCM, σκεφτείτε το εξής: Αν θέλουμε να έχουμε 256 επίπεδα ρύθμισης, ο κώδικας του BCM δεν χρειάζεται να επαναλαμβάνεται 256 φορές σε κάθε περίοδο αλλά μόλις 8. Έτσι, σε σχέση με ένα σύστημα PWM των 256 επιπέδων, το αντίστοιχο BCM είναι 32 φορές πιο αποδοτικό! Αν το PWM σπαταλάει το 30% του χρόνου του επεξεργαστή, το BCM απαιτεί λιγότερο από το 1%.



Εικόνα 1
Η μέση τιμή της τάσης του παλμού δεν καθορίζεται από τη συχνότητά του, ούτε από το πλήθος των μεταβολών στη διάρκεια μιας περιόδου. Τα μόνο μέγεθος που επιδρά στη μέση τιμή της τάσης είναι το Duty Cycle! Οι παλμοί στα τρία διαγράμματα, αν και παρουσιάζουν διαφορετική μορφή, διατηρούν σταθερό το Duty Cycle και ως εκ τούτου παρουσιάζουν την ίδια μέση τάση...

3. Το πλήθος των επιτρεπόμενων επιπέδων ρύθμισης.
4. Όσο αυξάνεται αυτή η συχνότητα, αυξάνεται και η σταθερότητα της παραγόμενης τάσης. Η συχνότητα του BCM παίζει τον ίδιο ρόλο με τη συχνότητα του PWM...

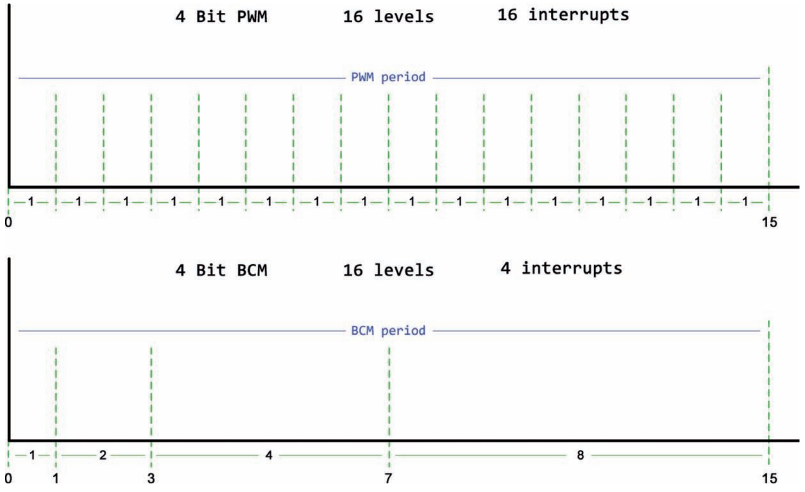
Η απαραίτητη επανάληψη

Πριν προχωρήσουμε στην αναλυτική παρουσίαση του BCM θα ήταν χρήσιμο να δώσουμε μια γενική περιγραφή: Η δημιουργία μιας τάσης με το Binary Code Modulation --όπως άλλωστε και με το Pulse Width Modulation-- επιτυγχάνεται με την ταχύτατη και κατάλληλη μεταβολή ενός ψηφιακού σήματος. Όπως αντιλαμβάνεστε, η ουσία του BCM κρύβεται πίσω από τη λέξη "κατάλληλη". Στη συνέχεια, λοιπόν, θα εξετάσουμε πώς μεταβάλλεται ένα σήμα BCM.

Για να κατανοήσει κανείς το BCM, πρέπει προηγουμένως να έχει κατανοήσει το PWM. Αν δεν έχετε διαβάσει το σχετικό άρθρο στο deltaHacker 007, θα σας προτείναμε να το κάνετε τώρα. Εδώ θα επαναλάβουμε μόνο το ρεζουμέ: Στο Pulse Width Modulation το σήμα είναι περιοδικό και η μέση τιμή της τάσης καθορίζεται από το Duty Cycle. Με άλλα λόγια, ο μέσος όρος της τάσης καθορίζεται από το πηλίκο του χρόνου κατά τον οποίο το σήμα βρίσκεται στη στάθμη HIGH (5Volt), προς τη διάρκεια της περιόδου του σήματος. Αν σας μπέρδεψαν αυτές οι περιγραφές, ρίξτε μια ματιά στο πρώτο διάγραμμα της εικόνας 1. Το Duty Cycle ισούται με το πηλίκο Ton / (Ton + Toff). Κατ' επέκταση, ο μέσος όρος της τάσης εξαρτάται από τους χρόνους Ton και Toff. Παρατηρήστε τώρα και τα υπόλοιπα διαγράμματα της εικόνας 1. Σε όλες τις περιπτώσεις, στη διάρκεια μιας περιόδους του σήματος τα δύο χρονικά μεγέθη (Ton και Toff) παραμένουν αμετάβλητα. Επομένως, το Duty Cycle παραμένει σταθερό και το ίδιο ισχύει και για τη μέση τιμή της τάσης. Το γεγονός ότι άλλαξε το "σχήμα" του παλμού δεν παίζει κανένα ρόλο. Μόνο το Duty Cycle επηρεάζει τον μέσο όρο της τάσης.

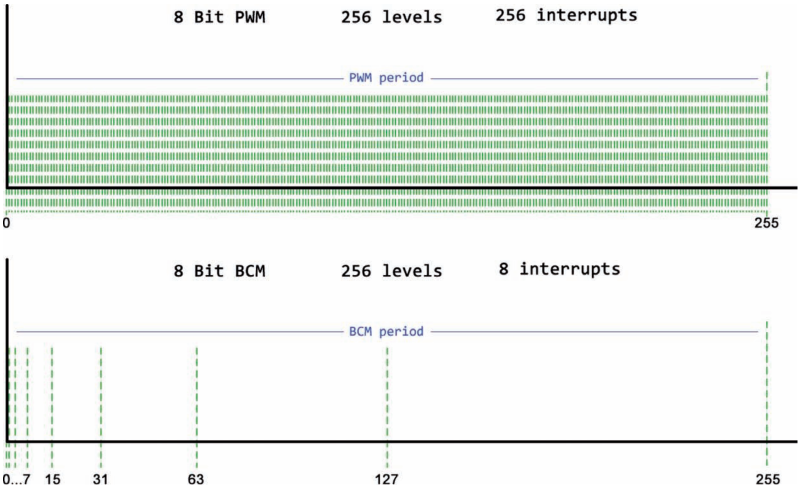
Από το PWM στο BCM

Όταν εμπεδώσει κανείς την παραπάνω παρατήρηση, αρχίζει να σκέφτεται πονηρά: Εφόσον το σχήμα του παλμού δεν επηρεάζει το μέσο όρο της τάσης, μπορούμε να το αλλάξουμε. Συγκεκριμένα, μπορούμε να το αλλάξουμε κατά τέτοιο τρόπο ώστε να σχηματίζεται ευκολότερα από το software. Όμως πώς σχηματίζεται ο παλμός στην περίπτωση του PWM; Για να αλλάξουμε κάτι, πρέπει να το γνωρίζουμε...



Εικόνα 2
Σε ένα 4μπιτο software PWM, η περίοδος του παλμού διαιρείται σε 16 περιοχές. Ο κώδικας που ελέγχει την κατάσταση του παλμού εκτελείται στην αρχή καθεμίας περιοχής, και κατ' επέκταση, εκτελείται 16 φορές ανά περίοδο. Σε ένα 4μπιτο BCM, όμως, η περίοδος διαιρείται μόνο σε 4 περιοχές. Επομένως, ο κώδικας που ελέγχει την κατάσταση του παλμού εκτελείται μόνο 4 φορές ανά περίοδο!

Εικόνα 3
Η διαφορά μεταξύ PWM και BCM αυξάνεται εκθετικά, όσο αυξάνονται τα bit ανάλυσης. Σε ένα 8μπιτο software PWM, ο σχετικός κώδικας εκτελείται 256 φορές ανά περίοδο. Σε ένα 8μπιτο BCM, όμως, ο κώδικας επαναλαμβάνεται μόλις 8 φορές!



Στο PWM, η περίοδος του παλμού χωρίζεται σε τόσα ίσα τμήματα όσα είναι τα διαθέσιμα επίπεδα ρύθμισης. Έτσι, για την υλοποίηση ενός 4μπιτου PWM (16 στάθμες ρύθμισης) η περίοδος του παλμού διαιρείται σε 16 ίσα τμήματα. Κάθε φορά που ξεκινά ένα τέτοιο τμήμα ο κώδικας ελέγχει αν ο παλμός βρίσκεται στην κατάσταση HIGH ή LOW και την προσαρμόζει αντίστοιχα. Ακριβώς γι' αυτό, στην υλοποίηση ενός 4μπιτου PWM ο σχετικός κώδικας εκτελείται 16 φορές ανά περίοδο[5]. Τώρα, πρέπει να επαναφέρουμε στη σκέψη μας το βασικό πόρισμα: Ο μέσος όρος της τάσης επηρεάζεται μόνο από το Duty Cycle. Εφόσον η κατάσταση του παλμού ελέγχεται πολλές φορές σε κάθε περίοδο, θα μπορούσε και να αλλάζει πολλές φορές ανά περίοδο. Στο PWM δεν συμβαίνει ποτέ κάτι τέτοιο, αλλά η θεωρία δεν το απαγορεύει! Σημασία έχει να πετυχαίνουμε ένα σταθερό Duty Cycle. Ας το κρατήσουμε αυτό κι ας προχωρήσουμε παρακάτω. Μάς υποχρεώνει κανείς να χωρίζουμε την περίοδο του παλμού σε ίσα τμήματα; Ξεχάστε για λίγο τους μετρητές, τους ελέγχους και κάθε άλλη λεπτομέρεια του κώδικα. Σκεφτείτε ότι αυτό που μας ενδιαφέρει κάθε φορά είναι να πετύχουμε ένα συγκεκριμένο Duty Cycle. Επομένως, υπάρχει μόνο μια προϋπόθεση για το πώς θα χωρίσουμε την περίοδο του παλμού: Τα επιμέρους τμήματα θα πρέπει να έχουν τέτοια μεγέθη και τέτοιο πλήθος, ώστε συνδυάζοντάς τα κατάλληλα να μπορούμε να πετύχουμε οποιοδήποτε Duty Cycle. Βέβαια σε ένα σύστημα PWM των 16 επιπέδων ρύθμισης, το Duty Cycle μπορεί να πάρει ακριβώς 16 τιμές (και όχι άπειρες). Κατά συνέπεια, στο σενάριο που εξετάζουμε αρκεί να χωρίσουμε την περίοδο του παλμού σε τόσα και τέτοια τμήματα, ώστε συνδυάζοντάς τα να μπορούμε να σχηματίσουμε 16 διαφορετικά Duty Cycle.

Όμορφα! Τώρα μπορούμε να συνοψίσουμε: Έχουμε έναν ψηφιακό παλμό (βρίσκεται πάντα σε μία από δύο πιθανές καταστάσεις) και θέλουμε να τον χωρίσουμε με κατάλληλο τρόπο, ώστε με τα επιμέρους τμήματα να μπορούμε να περιγράψουμε 16 διαφορετικά Duty Cycle. Πάει κάπου το μυαλό σας; Μα, φυσικά, στο δυαδικό σύστημα αρίθμησης και στον τρόπο με τον οποίο μεταβάλλεται η αξία κάθε ψηφίου! Αν πάντως το μυαλό σας δεν πήγε εκεί, μην ανησυχείτε. Πριν γνωρίσουμε το BCM, ούτε το δικό μας πήγε ;)

5. Αντίστοιχα, στην περίπτωση ενός 8μπιτου software PWM (256 στάθμες ρύθμισης) ο σχετικός κώδικας εκτελείται 256 φορές.

Σε αυτό το σημείο πρέπει να έχετε αρκετές απορίες. Η εικόνα 2 θα σας διαφωτίσει. Στο πάνω διάγραμμα φαίνεται ο τεμαχισμός της περιόδου στο Pulse Width Modulation, ενώ στο κάτω φαίνεται ο τεμαχισμός της περιόδου στο Binary Code Modulation. Στην περίπτωση του BCM η περίοδος του παλμού χωρίζεται σε τμήματα, καθένα από τα οποία έχει τη διπλάσια διάρκεια από το προηγούμενο. Αυτά τα τμήματα είναι μόλις τέσσερα. Δηλαδή, ακριβώς όσα δυαδικά ψηφία χρειάζονται για την περιγραφή 16 διαφορετικών μεγεθών. Στην εικόνα 4 φαίνεται ο παλμός στο 4μπιτο PWM, όταν η στάθμη έχει ρυθμιστεί στο (1011)2. Στην ίδια εικόνα φαίνεται και ο αντίστοιχος παλμός ενός 4μπιτου BCM. Τα βέλη δείχνουν την αντιστοιχία μεταξύ των ψηφίων της επιλεγμένης στάθμης (στο δυαδικό) και των τμημάτων της περιόδου.

Σημεία προσοχής

Η πλήρης κατανόηση του Binary Code Modulation, τουλάχιστον στην αρχή φαντάζει δύσκολη. Ωστόσο πρόκειται για πανίσχυρο εργαλείο και, πραγματικά, πρέπει να επιμείνετε. Στη συνέχεια παραθέτουμε τα γενικότερα χαρακτηριστικά ενός συστήματος BCM.

- Σε μια 4μπιτη υλοποίηση του BCM η περίοδος του παλμού χωρίζεται σε τέσσερα τμήματα. Αντίστοιχα, σε μια 8μπιτη υλοποίηση η περίοδος χωρίζεται σε οκτώ τμήματα. Το γιατί χωρίζουμε την περίοδο σε τόσα τμήματα όσα είναι και τα bit, θα φανεί παρακάτω.
- Η πρώτη περιοχή έχει τη μικρότερη διάρκεια. Η δεύτερη έχει την ακριβώς διπλάσια, η τρίτη την τετραπλάσια και πάει λέγοντας. Γενικότερα, καθεμία από τις περιοχές (στις οποίες χωρίζεται η περίοδος του παλμού) έχει τη διπλάσια διάρκεια σε σχέση με την αμέσως προηγούμενη.
- Η διάρκεια της πρώτης περιοχής υπολογίζεται εύκολα. Αρκεί να μειώσουμε κατά μια μονάδα το πλήθος των επιπέδων ρύθμισης και να διαιρέσουμε με αυτόν τον αριθμό την περίοδο του παλμού. Για παράδειγμα, σε ένα 4μπιτο BCM (16 επίπεδα ρύθμισης) η πρώτη περιοχή του παλμού διαρκεί όσο το 1/15 της περιόδου[6].
- Προφανώς, σε μία υλοποίηση του BCM των k bit, η στάθμη στην οποία μπορούμε να ρυθμίσουμε το σύστημα περιγράφεται από έναν δυαδικό αριθμό με k στο πλήθος ψηφία. Το πρώτο ψηφίο αυτού του αριθμού (το ψηφίο με τη μικρότερη αξία) αντιστοιχεί στην πρώτη περιοχή της περιόδου (εκείνη με τη μικρότερη διάρκεια). Το δεύτερο ψηφίο αντιστοιχεί στη δεύτερη περιοχή κ.ο.κ. Η τιμή κάθε ψηφίου δηλώνει την κατάσταση στην οποία βρίσκεται ο παλμός στην αντίστοιχη περιοχή.
- Ο έλεγχος για την κατάσταση του παλμού (και η ενδεχόμενη αλλαγή του) πραγματοποιείται τόσες φορές όσες είναι και οι περιοχές στις οποίες έχει διαιρεθεί η περίοδος. Επομένως, σε ένα 4μπιτο BCM ο σχετικός έλεγχος πραγματοποιείται 4 φορές σε κάθε περίοδο του παλμού. Γενικά, σε μια υλοποίηση του BCM των k bit, ο κώδικας πρέπει να ελέγχει κι ενδεχομένως να μεταβάλλει την κατάσταση του παλμού k φορές ανά περίοδο.

6. Η δεύτερη περιοχή διαρκεί 2/15 της περιόδου, η τρίτη διαρκεί 4/15 της περιόδου και η τέταρτη διαρκεί 8/15 της περιόδου. Όλες μαζί διαρκούν (1+2+4+8)/15 = 1 περίοδο!

Περισσότερη οικονομία

Ο επεξεργαστής του Arduino επιτρέπει την αυθαίρετη ενεργοποίηση και απενεργοποίηση ορισμένων κυκλωμάτων του. Έτσι, όταν δεν χρησιμοποιούμε ορισμένα περιφερειακά μπορούμε να τα απενεργοποιούμε και να εξοικονομούμε ενέργεια συνεχώς -- ακόμη και όταν ο επεξεργαστής δεν κοιμάται! Σε αυτή την ομάδα περιφερειακών ανήκουν ο μετατροπέας αναλογικού σήματος σε ψηφιακό (Analog to Digital Converter - ADC) και ο συγκριτής τάσεων (Analog Comparator - AC). Επομένως, όταν δεν σκοπεύουμε να χρησιμοποιήσουμε τις αναλογικές εισόδους του Arduino, μπορούμε να εξοικονομήσουμε ενέργεια απενεργοποιώντας το κύκλωμα ADC. Για το σκοπό αυτό, αρκεί να θέσουμε σε 0 το bit ονόματι ADEN, του καταχωρητή ADCSRA. Αυτό μπορεί να γίνει εύκολα με διάφορες πράξεις bitwise, αλλά εφόσον θέλουμε να απενεργοποιήσουμε τον ADC (γεγονός που σημαίνει ότι δεν μας απασχολούν οι ρυθμίσεις του), μπορούμε πολύ απλά να δώσουμε το εξής:

```
ADCSRA = 0;
```

Η χρήση του συγκριτή τάσεων είναι πιο προχωρημένη. Αν δεν ξέρετε τι κάνει, τότε σίγουρα δεν τον χρειάζεστε και μπορείτε να τον απενεργοποιήσετε. Για το σκοπό αυτό αρκεί να θέσετε σε 1 το bit ονόματι ACD, στον καταχωρητή ACSR. Κάτι τέτοιο μπορεί να γίνει πανεύκολα, ως εξής:

```
ACSR = 255;
```

Παρεμπιπτόντως, έχετε υπόψη ότι μπορούμε να απενεργοποιήσουμε και πρόσθετα υποσυστήματα, με τη βοήθεια του καταχωρητή PRR (Power Reduction Register). Ωστόσο, η χρήση του απαιτεί λίγη προσοχή, καθώς είναι εύκολο να απενεργοποιήσουμε συστήματα που δεν θα έπρεπε, με απρόβλεπτες συνέπειες για το πρόγραμμά μας. Μπορείτε να μάθετε τα πάντα για τον PRR, από τις σελίδες 43 και 46, στο datasheet του επεξεργαστή.

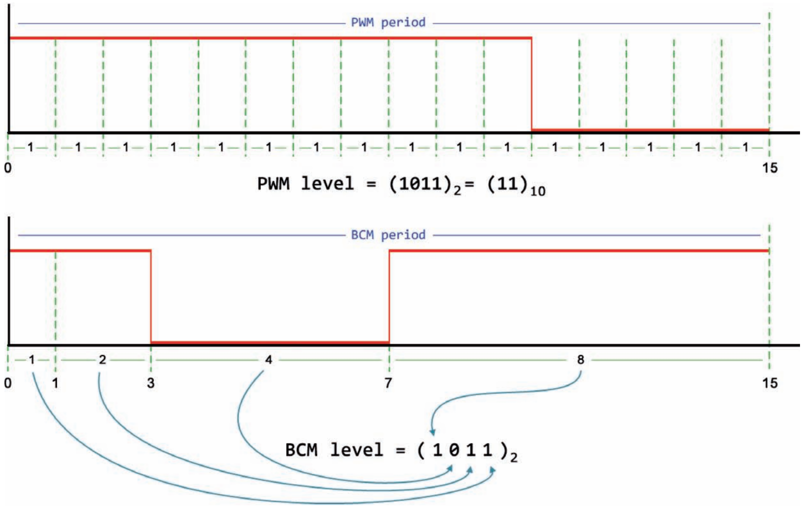
Τέλος, πρέπει να σας πούμε ότι η πλακέτα του Arduino περιλαμβάνει έναν ενεργοβόρο σταθεροποιητή τάσης. Επιπρόσθετα, το τσιπάκι για την επικοινωνία μέσω USB --στις σύγχρονες εκδοχές του Arduino UNO-- είναι επίσης σπάταλο. Σε αυτές τις πλακέτες, βλέπετε, η επικοινωνία επιτυγχάνεται με τη βοήθεια ενός επεξεργαστή, ισοδύναμου με τον κεντρικό του Arduino! Σαν συνέπεια των παραπάνω, οτιδήποτε κι αν κάνουμε στον κώδικα του δικού μας προγράμματος είναι σχεδόν αδύνατο να μειώσουμε τη συνολική κατανάλωση της πλακέτας. Πάντως σε επόμενο άρθρο θα μάθουμε πώς να μεταφέρουμε τα προγράμματά μας και τον επεξεργαστή *εκτός* της πλακέτας του Arduino. Τότε, όλα όσα μάθαμε σήμερα θα αποδειχθούν πολύτιμα.

Στην πράξη

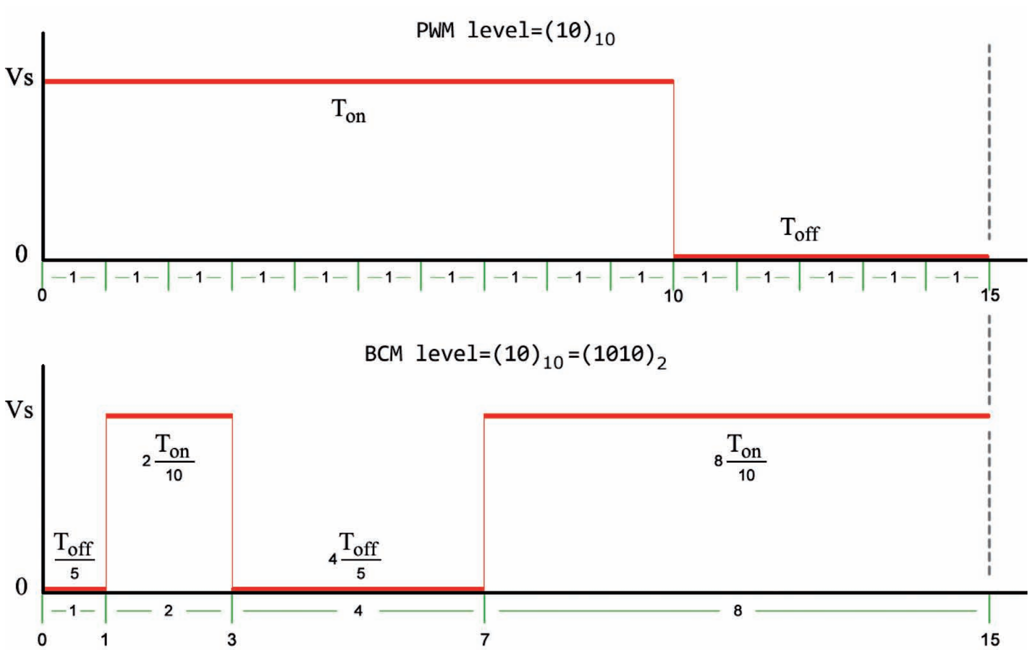
Εάν η θεωρία δεν σας βοήθησε να ξεκαθαρίσετε τα πάντα, πιστεύουμε ότι θα το πετύχει ο κώδικας. Πριν προχωρήσουμε όμως θα σας προτείναμε να κατεβάσετε το πακέτο bcmsrc.zip από το `http://bit.ly/dh009bcmsrc`. Σε αυτό θα βρείτε δύο υλοποιήσεις του Binary Code Modulation, καθώς και μια μικρή εφαρμογή. Όλα τα προγραμματάκια προϋποθέτουν το απλό κύκλωμα των δέκα LED, το οποίο συναντήσαμε και στην παρουσίαση

του software PWM (εικόνα 6). Η πρώτη υλοποίηση του BCM είναι 4μπιτη και πραγματοποιεί επαναλαμβανόμενα ένα fade-in στα μισά LED και ένα fade-out στα άλλα μισά. Το ίδιο κάνει και η δεύτερη υλοποίηση του BCM, η οποία είναι 8μπιτη. Αν υλοποιήσετε το απλό κύκλωμα και φορτώσετε τα δύο προγράμματα, θα διαπιστώσετε ότι το fading στην 8μπιτη υλοποίηση είναι αισθητά πιο ομαλό! Στη δεύτερη υλοποίηση βασίζεται και το τρίτο πρόγραμμα, το οποίο χρησιμοποιεί τα δέκα LED για την αναπαράσταση ενός αριθμού στο δυαδικό. Αυτός ο αριθμός αυξάνεται με την πάροδο του χρόνου, αλλά τα LED μεταβαίνουν από τη μία κατάσταση στην επόμενη με το εφέ του morphing. Παρακάτω θα εξετάσουμε τα πιο ενδιαφέροντα τμήματα του προγράμματος.

```
ISR(TIMER2_COMPA_vect)
{
      OCR2A = bit_mask;
      portd_data = 0;
      portb_data = 0;
      if (levels[0] & bit_mask) {portd_data |= 1;}
      […]
      if (levels[9] & bit_mask) {portb_data |= 2;}
      PORTD = portd_data;
      PORTB = portb_data;
      bit_mask = bit_mask * 2;
      if (bit_mask == 0) {
                bit_mask = 1;
                bcm_cycles++;
      }
}
```



Εικόνα 4
Στο Binary Code Modulation, κάθε περιοχή στην οποία χωρίζεται η περίοδος του παλμού αντιστοιχεί σε διαφορετικό bit του αριθμού που περιγράφει την επιλεγμένη στάθμη. Ακριβώς γι' αυτό, κάθε περιοχή έχει διπλάσιο μέγεθος από την προηγούμενη και το μισό από την επόμενη -- όπως συμβαίνει με τις αξίες των ψηφίων στο δυαδικό σύστημα. (Σ.τ.Ε. Πλεονασμός :D #math_troll)



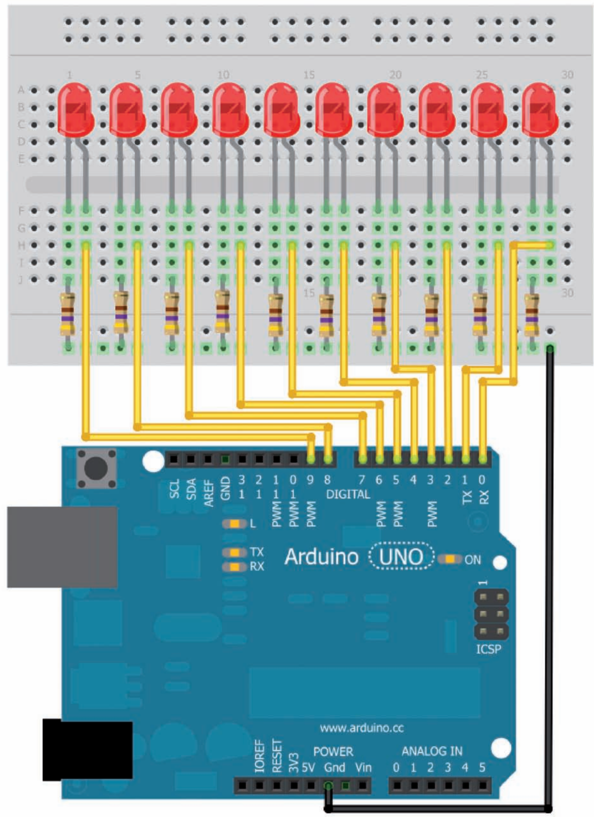
Εικόνα 5
Το σύστημα είναι ρυθμισμένο στη στάθμη (1010)2. Έτσι, στην περίπτωση του BCM ο παλμός βρίσκεται στην κατάσταση HIGH σε δύο περιοχές: Στη δεύτερη και στην τέταρτη. Η αντιστοιχία μεταξύ ψηφίων στο δυαδικό και περιοχών της περιόδου, είναι ολοφάνερη!

Πρόκειται για τη ρουτίνα εξυπηρέτησης του interrupt, το οποίο προκύπτει όταν ο Timer2 φτάνει στην τιμή του OCR2A. Εδώ βρίσκεται ο κώδικας που σχηματίζει έναν παλμό BCM, για καθένα από τα δέκα LED. Πριν πούμε οτιδήποτε άλλο, σημειώστε ότι ο Timer2 είναι ρυθμισμένος ώστε να υπερχειλίζει και να ξεκινά πάλι από το μηδέν (CTC mode) κάθε φορά που φτάνει στην τιμή του OCR2A. Επίσης, έχετε υπόψη ότι η στάθμη για κάθε LED είναι αποθηκευμένη στον πίνακα levels.

Όταν ξεκινά το πρόγραμμα, η μεταβλητή bit_mask αρχικοποιείται με την τιμή 1. Έτσι, την πρώτη φορά που εκτελείται το παραπάνω απόσπασμα ο καταχωρητής OCR2A αποκτά την τιμή 1. Αυτό σημαίνει ότι ο Timer2 θα παράγει το επόμενο interrupt όταν φτάσει στην τιμή 1. Ακριβώς τότε, θα ξεκινήσει εκ νέου η εκτέλεση του συγκεκριμένου τμήματος κώδικα. Ουσιαστικά, αποδίδοντας αυτήν την τιμή στον OCR2A ορίζουμε την πρώτη (και μικρότερη) περιοχή της περιόδου του BCM. Επομένως ο κώδικάς μας πρέπει να διαμορφώσει τον παλμό για κάθε LED, σύμφωνα με την τιμή του πρώτου bit της στάθμης καθενός LED. Αυτό επιτυγχάνεται με τη λογική πράξη AND, μεταξύ της στάθμης του εκάστοτε LED και της μεταβλητής bit_mask. Αν το αποτέλεσμα είναι διάφορο του μηδενός, το πρώτο bit της στάθμης ήταν μονάδα και η αντίστοιχη έξοδος τίθεται στο λογικό HIGH. Όταν ολοκληρωθούν οι έλεγχοι, ο κώδικας ενημερώνει όλους τους ακροδέκτες εξόδου ταυτόχρονα και διπλασιάζει την τιμή της bit_mask. Με αυτόν το διπλασιασμό η μεταβλητή bit_mask προετοιμάζεται για την επόμενη εκτέλεση της συγκεκριμένης ρουτίνας. Βλέπετε, κάθε φορά που εκτελείται η ρουτίνα, ο καταχωρητής OCR2A πρέπει να αποκτά τη διπλάσια τιμή ώστε να οριοθετεί μια νέα περιοχή της περιόδου του BCM με τη διπλάσια διάρκεια[7]. Εξάλλου, διπλασιάζοντας κάθε φορά την bit_mask μπορούμε να ελέγξουμε την τιμή του επόμενου bit της στάθμης

7. Όπως είδαμε στη θεωρία, κάθε περιοχή της περιόδου του BCM, πρέπει να έχει τη διπλάσια διάρκεια από την αμέσως προηγούμενη.

καθενός LED. Όπως αντιλαμβάνεστε, με τους διαδοχικούς διπλασιασμούς η τιμή bit_mask θα φτάσει κάποια στιγμή στην τιμή 128. Έχοντας αυτήν την τιμή η επόμενη εκτέλεση του κώδικα θα ορίσει την όγδοη και μεγαλύτερη περιοχή της περιόδου του BCM, ενώ θα ελέγξει το όγδοο bit της στάθμης για κάθε LED. Πλέον, ο διπλασιασμός τη bit_mask θα της αποδώσει την τιμή 0[8]. Εκεί παρεμβαίνει ο τελευταίος έλεγχος που εμφανίζεται στο παραπάνω απόσπασμα. Εάν η τιμής της bit_mask μηδενιστεί, θα πρέπει να αποκτήσει την τιμή 1. Βλέπετε, μετά την όγδοη περιοχή την οποία σηματοδότησε η τιμή 128 της bit_mask, η περίοδος του BCM έχει ολοκληρωθεί! Στην επόμενη εκτέλεση του κώδικα θα πρέπει να ξεκινήσει η πρώτη και μικρότερη περιοχή της νέας περιόδου. Μαζί με την απόδοση της τιμής 1 στη bit_mask αυξάνεται και τιμή του μετρητή bcm_cycles. Προφανώς, ο συγκεκριμένος μετρητής απαριθμεί το πλήθος των ολοκληρωμένων περιόδων του BCM. Η τιμή αυτού του μετρητή αξιοποιείται σε άλλες περιοχές του προγράμματος.



Εικόνα 6
Το κύκλωμα που προϋποθέτουν τα προγράμματά μας. Πρόκειται για το απλούστατο κύκλωμα που συναντήσαμε και στη μελέτη του software PWM.

```
void loop () {
    if (bcm_cycles == 4) {
        bcm_cycles = 0;
        morphing();
        if (diff_cnt == 0) { new_state(); }
    }
    sleep_mode();
}
```

Ο κύριος βρόχος του προγράμματος αποτελεί το αμέσως πιο ενδιαφέρον τμήμα κώδικα, μετά από εκείνο που υλοποιεί το Binary Code Modulation. Εδώ, η λογική του προγράμματος έχει ως εξής: Όταν ολοκληρωθούν τέσσερεις περίοδοι του BCM (bcm_cycles==4), μηδενίζεται ο σχετικός μετρητής και εκτελείται η ρουτίνα morphing. Ο μετρητής bcm_cycles, δηλαδή, χρησιμοποιείται μόνο και μόνο για να εισαγάγει μια καθυστέρηση στην εξέλιξη του προγράμματος. Όμως τι κάνει η ρουτίνα morphing;

Όπως είπαμε, το πρόγραμμά μας υλοποιεί έναν μετρητή, την τιμή του οποίου αναπαριστούν τα LED (στο δυαδικό). Η συνάρτηση morphing πε-

8. Η συγκεκριμένη μεταβλητή έχει οριστεί σαν Byte. Επομένως, η μέγιστη τιμή στην οποία μπορεί να φτάσει είναι το 255. Η εκτέλεση οποιασδήποτε πράξης, η οποία θα είχε σαν αποτέλεσμα το 256, έχει σαν συνέπεια την υπερχείλιση της μεταβλητής και συνεπώς το μηδενισμό της.

τυχαίνει την ομαλή/σταδιακή μετάβαση των LED, από τη μία τιμή του μετρητή στην επόμενη. Για τη λειτουργία αυτού του μηχανισμού, έχουμε ορίσει δύο πίνακες: Ο πρώτος (levels) περιλαμβάνει τις καταστάσεις των LED που περιγράφουν την τρέχουσα τιμή του μετρητή. Ο δεύτερος πίνακας (new_levels) περιέχει τις καταστάσεις των LED που περιγράφουν την επόμενη τιμή του μετρητή. Όταν ένα LED πρέπει να ανάψει, η αντίστοιχη τιμή στον πίνακα new_levels είναι 255. Αντίθετα, όταν πρέπει να σβήσει, η αντίστοιχη τιμή στον πίνακα new_levels είναι 0. Με την ίδια λογική, όταν ένα LED είναι αναμμένο η αντίστοιχη τιμή στον πίνακα levels είναι 255, ενώ όταν είναι σβηστό η αντίστοιχη τιμή είναι μηδέν. Ας υποθέσουμε, λοιπόν, ότι ένα LED είναι σβηστό για την τρέχουσα τιμή του μετρητή, αλλά πρέπει να ανάψει για την επόμενη. Η συνάρτηση morphing εξετάζει αρχικά τις τιμές που περιγράφουν την τρέχουσα (πίνακας levels) και την επόμενη κατάσταση (new_levels) του LED. Εφόσον η τιμή της επόμενης κατάστασης είναι μεγαλύτερη, η συνάρτηση θα αυξήσει την τιμή στον πίνακα levels. Αυτή η αύξηση θα επαναλαμβάνεται σε κάθε εκτέλεση της morphing, έως ότου η τιμή που περιγράφει την τρέχουσα κατάσταση του LED εξισωθεί με εκείνη που περιγράφει την επόμενη. Με αυτόν τον τρόπο, η κατάσταση του LED όπως περιγράφεται στον πίνακα levels, θα περάσει σταδιακά από την τιμή 0 στην τιμή 255. Κατά συνέπεια, το LED θα ανάψει βαθμιαία και ομαλά! Όπως αντιλαμβάνεστε, αν το LED έπρεπε να σβήσει η συνάρτηση morphing θα μείωνε την τιμή που περιγράφει την τρέχουσα κατάσταση έως ότου μηδενιζόταν. Έτσι, το LED θα έσβηνε ομαλά. Εκτός από αυτές τις μεταβολές στις τιμές του πίνακα levels, η morphing απαριθμεί και τις αλλαγές που πραγματοποίησε σε κάθε εκτέλεσή της. Αυτό το πλήθος αποθηκεύεται στη μεταβλητή diff_cnt. Ας επιστρέψουμε τώρα στον κύριο βρόχο του προγράμματός μας.

Μετά την κλήση της morphing, ο κώδικας ελέγχει την τιμή της diff_cnt. Αν είναι μηδέν, η morphing δεν χρειάστηκε να κάνει καμία αλλαγή. Αυτό σημαίνει ότι οι τιμές του πίνακα levels έχουν εξισωθεί με τις αντίστοιχες τιμές του πίνακα new_levels. Με άλλα λόγια, τα LED έχουν μεταβεί πλέον στην επόμενη κατάσταση και αναπαριστούν τη νέα τιμή του μετρητή! Ακριβώς γι’ αυτό καλείται η συνάρτηση new_state. Αυτή η συνάρτηση αυξάνει το μετρητή του προγράμματος και, σύμφωνα με τη νέα του τιμή, συμπληρώνει τον πίνακα new_levels. Έτσι, τα LED πρέπει να ξεκινήσουν μια νέα μετάβαση -- προς την κατάσταση που θα περιγράφει τη νέα τιμή του μετρητή.

Τέλος, θυμηθείτε ότι όσα περιγράψαμε συμβαίνουν μόνο κατά την ολοκλήρωση τεσσάρων περιόδων του BCM (bcm_cycles==4). Ωστόσο, ο κύριος βρόχος του προγράμματος εκτελείται διαρκώς. Σε κάθε επανάληψή του εκτελείται η sleep_mode() και ο επεξεργαστής του Arduino κοιμάται. Από αυτόν τον ύπνο ξυπνά από το interrupt, το οποίο σηματοδοτεί την αρχή μιας νέας περιοχής της περιόδου του BCM. Μόλις ολοκληρωθεί ο σχετικός κώδικας το πρόγραμμα επιστρέφει στον κύριο βρόχο, ελέγχει αν έχουν ολοκληρωθεί 4 περίοδοι του BCM κι αν όχι ο επεξεργαστής κοιμάται ξανά!



Skill: Beginner

Tags: Bruteforce, online password cracking, wordlist, md5 hash

ΕΠΙΤΕΘΕΙΤΕ ΜΕ ΟΟΟΟΟΛΗ ΣΑΣ ΤΗ ΔΥΝΑΜΗ

Οι δουλειές δεν θέλουν κόπο, θέλουν τρόπο, λέει η λαϊκή θυμοσοφία. Όταν όμως δεν υπάρχει ο τρόπος, δεν γλυτώνουμε τον κόπο. Επίσης, οι δουλειές θέλουν υπομονή κι επιμονή, αλλά δεν σκοπεύουμε να σας αραδιάσουμε όλα τα κλισέ. Θα μιλήσουμε μόνο για τον πρωτόγονο τρόπο, με τον οποίο γίνονται όλες οι φαινομενικά αδύνατες δουλειές: Την εφαρμογή όλης μας της δύναμης!

by Dr@G

Μην ανησυχείτε. Δεν θα σας προτείνουμε να κοπανήσετε το πληκτρολόγιο. Στην περίπτωση που εξετάζουμε, όταν μιλάμε για εφαρμογή δύναμης εννοούμε την επίμονη και σχεδόν τυφλή επανάληψη μιας διαδικασίας -- μέχρι να τα καταφέρουμε. Φανταστείτε ένα πολύπλοκο μηχανικό κατασκεύασμα, το οποίο έχει φρακάρει και δεν έχετε απολύτως καμία ιδέα για το πώς θα ξεκολλήσει. Μπροστά σε μια τέτοια κατάσταση, το πιο πιθανό θα ήταν να πάρετε ένα εργαλείο και ν' αρχίσετε να πιέζετε το μηχανισμό, μέχρι να ξεκολλήσει ή να κουραστείτε ;) Καλά όλα αυτά, θα πείτε, αλλά τι σχέση έχουν με τα δικά μας; Κάτι μάς λέει ότι αν πάψουμε να μιλάμε γενικά και δώσουμε μερικά παραδείγματα, θα καταλάβετε αμέσως.

Φανταστείτε, λοιπόν, ένα σύστημα το οποίο δεν έχει καμία φανερή αδυναμία (vulnerability) και για το οποίο γνωρίζετε μόνο το username κάποιου χρήστη. Αν θελήσετε να συνδεθείτε, από τη στιγμή που δεν γνωρίζετε το password θα πρέπει να καταφύγετε στη "βάρβαρη" λύση της διαδοχικής δοκιμής *όλων* των πιθανών passwords! Έτσι, κάααποια στιγμή θα βρείτε το σωστό password και θα συνδεθείτε -- ή θα κουραστείτε και θα παραιτηθείτε. Νομίζουμε ότι ο παραλληλισμός με το μηχανικό σύστημα είναι πλέον προφανής. Μπροστά σ' ένα πρόβλημα, το οποίο δεν λύνεται με κανένα τέχνασμα, η μόνη λύση έγκειται στις ατελείωτες και υπομονετικές δοκιμές. Ουσιαστικά, αυτή η προσέγγιση αποτελεί την απλούστερη και παλαιότερη μέθοδο επίθεσης, η οποία ακούει στο όνομα bruteforce attack.

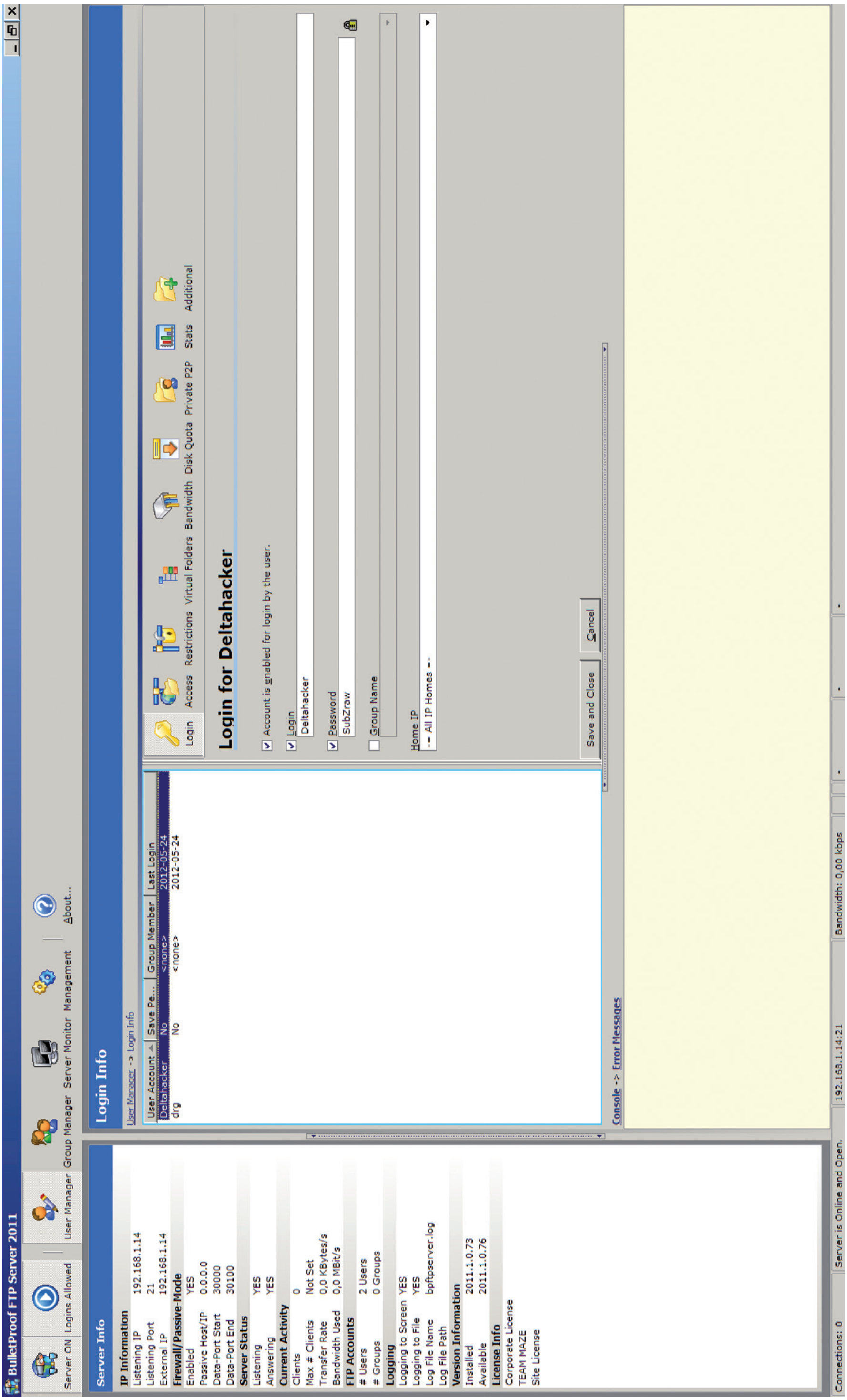
Η τεχνική του bruteforce είναι παλιά, βάρβαρη και εντελώς απλοϊκή. Αυτό όμως δεν σημαίνει ότι θα καταπιαστούμε με κάποια επίπονη διαδικασία, ούτε ότι θα σπαταλήσουμε όλο μας το χρόνο κάνοντας τα ίδια και τα ίδια. Εφόσον βρισκόμαστε μπροστά σε υπολογιστή, θα τον αφήσουμε να κάνει αυτός τη βρώμικη δουλειά! Καταρχάς, όταν εξαπολύουμε μια επίθεση bruteforce πρέπει να εξασφαλίσουμε ότι θα ολοκληρωθεί το ταχύτερο δυνατό. Γι' αυτό το λόγο, οι δοκιμές πρέπει να ξεκινούν με τα πιο πιθανά συνθηματικά. Αναρωτιέστε που θα βρούμε τα πιθανά συνθηματικά ή πώς θα τα μαντέψουμε; Αν ναι, θα πρέπει να διαβάσετε τα άρθρα για το wordlist generation, στα τεύχη 005 και 006.

Στη συνέχεια θα εξετάσουμε μια κλασσική περίπτωση bruteforce attack και θα δούμε ένα script γραμμένο σε Python, το οποίο αυτοματοποιεί τις επιθέσεις του είδους. "Let the game begin", όπως θα έλεγε κι ο Jigsaw από το SAW! (Σ.τ.Ε. Από τα ελάχιστα θρίλερ που με ανατριχιάζουν. Αλήθεια λέω.)

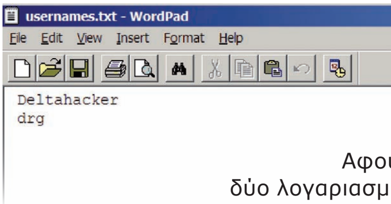
Bruteforce attack με την Python

Για τις δοκιμές μας δεν θα επιτεθούμε σε κάποιον αληθινό server. Θα ήταν άσκοπο κι ενοχλητικό για το ξένο μηχάνημα, θα συνιστούσε ανώριμη συμπεριφορά και πιθανώς να μας έβαζε και σε μπελάδες. Αντίθετα, θα εγκαταστήσουμε στο δικό μας μηχάνημα μια δικτυακή υπηρεσία και θα κάνουμε τις δοκιμές μας πάνω της. Συγκεκριμένα, θα εγκαταστήσουμε έναν FTP server και θα εξετάσουμε ένα script, το οποίο αυτοματοποιεί μια επίθεση bruteforce.

Το σύστημα της δοκιμής έτρεχε Windows και για τις ανάγκες μας εγκαταστήσαμε τον εμπορικό FTP server ονόματι BulletProof (`www.bpftpserver.com`). Φυσικά, εσείς μπορείτε να εγκαταστήσετε οποιονδήποτε άλλον θέλετε. Σε κάθε περίπτωση, στο μηχάνημα από το οποίο θα κάνετε την επίθεση, πρέπει να εγκατασταθεί η έκδοση 2.7.2 της Python. Αν χρησιμοποιείτε Windows (όπως εμείς) μπορείτε να την κατεβάσετε από εδώ:



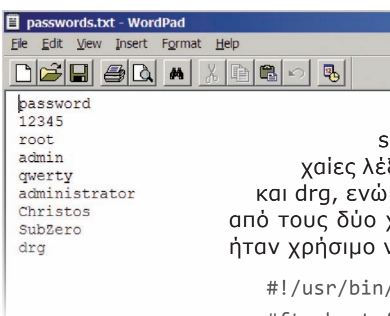
Εικόνα 1
Ο BulletProof FTP server είναι απλός στη χρήση και προσφέρει αρκετές ευκολίες. Γενικότερα, πάντως, γνώμη του γράφοντα είναι πως ένα σύστημα με Windows δεν αποτελεί τον ιδανικό server. Αποτελεί ωστόσο τον ιδανικό στόχο για πειραματισμούς...



Εικόνα 2
Το αρχείο usernames.txt περιλαμβάνει μόνο τα usernames των χρηστών DeltaHacker και drg.

http://www.python.org/ftp/python/2.7.2/python-2.7.2.msi. Αν χρησιμοποιείτε Linux/BSD τότε είτε είναι ήδη εγκατεστημένη είτε μπορείτε να την εγκαταστήσετε εύκολα, καταφεύγοντας στο σχετικό εργαλείο διαχείρισης πακέτων.

Αφού εγκατασταθεί ο FTP server, πρέπει να δημιουργήσουμε δύο λογαριασμούς χρήστη. Ο ένας θα είχε username το drg και password το drg, ενώ ο άλλος DeltaHacker και SubZero αντίστοιχα. Για τη δημιουργία των λογαριασμών, αρκεί να κάνουμε κλικ στο κουμπί User Manager (πάνω αριστερά). Από την περιοχή στην οποία θα μεταφερθούμε, πιστεύουμε ότι δεν θα δυσκολευτείτε καθόλου να προσθέσετε τους δύο χρήστες (add user) και να τροποποιήσετε τις ρυθμίσεις του καθενός (edit user). Αμέσως μετά πρέπει να επιβεβαιώσουμε ότι έχει ξεκινήσει η υπηρεσία καθώς κι ότι επιτρέπει τις εισερχόμενες συνδέσεις. Αρκεί μια ματιά στα πρώτα δύο κουμπιά, στο πάνω αριστερό άκρο του μενού του προγράμματος. Αν κάποιο από τα δύο δεν είναι πατημένο, ε, το πατάμε. Πλέον, μπορούμε να εστιάσουμε στην επίθεση καθεαυτή, χρησιμοποιώντας το script που φιλοξενείται εδώ: http://pypi.python.org/pypi/ftp_bruteforce/1.0.0. Το συγκεκριμένο script αναλαμβάνει το ρόλο ενός FTP client, ο οποίος πραγματοποιεί αλλεπάλληλες απόπειρες σύνδεσης δοκιμάζοντας διάφορους συνδυασμούς username και password. Με απλά λόγια, το συγκεκριμένο script πραγματοποιεί μια επίθεση bruteforce στον FTP server που του υποδεικνύουμε. Όμως για να λειτουργήσει το script χρειάζεται και δύο βοηθητικά αρχεία, απλού κειμένου. Το ένα περιλαμβάνει τα πιθανά usernames και το άλλο τα πιθανά passwords. Αν αναρωτιέστε πώς θα κατασκευάσετε αυτά τα αρχεία, θα σας το ξαναπούμε: Διαβάστε τα σχετικά άρθρα στα τεύχη 005 και 006 ;) Πάντως, για μια απλή δοκιμή του script, μπορούμε να φτιάξουμε δύο αρχεία με μερικές τυχαίες λέξεις. Στο ένα θα πρέπει να συμπεριλάβουμε τα DeltaHacker και drg, ενώ στο άλλο θα πρέπει να συμπεριλάβουμε το password ενός από τους δύο χρήστες. Πριν δούμε πώς συνδυάζονται τα παραπάνω, θα ήταν χρήσιμο να κατανοήσουμε τη λειτουργία του script.



Εικόνα 3
Το αρχείο passwords.txt περιέχει τα πιθανά passwords. Σε ένα ρεαλιστικό σενάριο, το εν λόγω αρχείο θα περιείχε ολόκληρα wordlists!

```
#!/usr/bin/python
#ftp_bruteforce.py by Dr@G
#python ftp_bruteforce.py <host> <users.txt> <pass.txt>
import socket
import re
import sys
#-Initial arguments,read usernames and passwords files------
ftp_domain=sys.argv[1]
usersfilename=sys.argv[2]
passwordsfilename=sys.argv[3]
u_file=open(usersfilename,'r')
p_file=open(passwordsfilename,'r')
#-----------Append file content to lists-------------------
user_file=[]
passwords_file=[]
for line in u_file:
  user_file.append(line[:-1])
```

```
for line in p_file:
  passwords_file.append(line[:-1])
#--------------Connect to FTP-server function---------------
def ftpconnect(username,password):
  s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
  print "Testing [User]: " + username + " [Password]: " + password
  s.connect((ftp_domain,21))
  data = s.recv(1024)
  s.send('USER ' + username + '\r\n')
  data = s.recv(1024)
  s.send('PASS ' + password + '\r\n')
  data = s.recv(3)
  s.send('QUIT\r\n')
  s.close()
  return data
#------------Bruteforce----------------------------------
def bruteforce():
  for user in user_file:
    for password in passwords_file:
      attempt=ftpconnect(user,password)
      if attempt == "230":
        print "-------------------------------------------"
        print "[Found] "+user+":"+password
        print "-------------------------------------------"
#-----------------------------------------------------------
bruteforce()
```

Η λογική του κώδικα

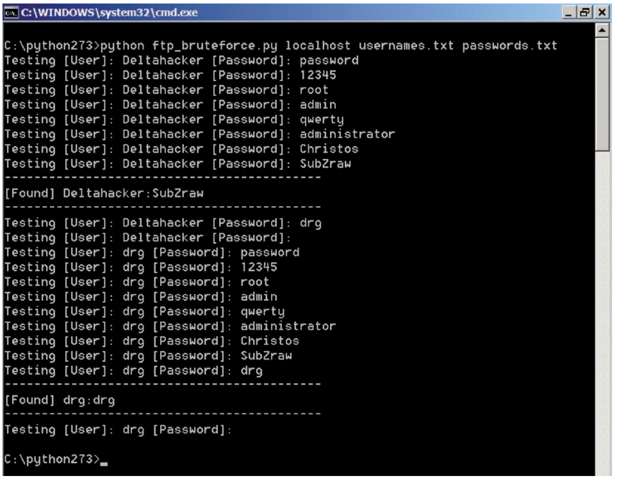
Το πρόγραμμα δέχεται από το χρήστη τρία ορίσματα: Το <host>, το <users.txt> και το <pass.txt>. Το πρώτο όρισμα αποτελεί τη διεύθυνση του στόχου μας. Το δεύτερο όρισμα αποτελεί το αρχείο με τα πιθανά usernames, ενώ το τρίτο υποδεικνύει το αρχείο με τα πιθανά passwords. Όλα τα σχετικά με την "ανάγνωση" των ορισμάτων και τη μεταφορά τους σε μεταβλητές του script, πραγματοποιούνται στο πρώτο τμήμα του κώδικα. Στο αμέσως επόμενο, τα περιεχόμενα των αρχείων μεταφέρονται σε δύο λίστες, απαλλαγμένες από τους χαρακτήρες αλλαγής γραμμής. Στο τρίτο τμήμα του script δημιουργούμε μια συνάρτηση που πραγματοποιεί τη σύνδεση με τον FTP server, δοκιμάζει τα username και password που της έχουμε δώσει σαν ορίσματα κι επιστρέφει την απάντηση του server. Στο τέταρτο και τελευταίο τμήμα του κώδικα βρίσκεται ένας βρόχος επαναλήψεων. Σ' αυτόν το βρόχο, για κάθε ζευγάρι username και password καλείται η συνάρτηση σύνδεσης στον FTP server. Αν το αποτέλεσμα της κλήσης είναι ίσο με το 230, το script συμπεραίνει ότι η σύνδεση ήταν επιτυχής και τυπώνει το "νικητήριο" συνδυασμό username και password.

Δοκιμή

Νομίζουμε ότι τώρα έχουν ξεκαθαρίσει τα πάντα. Ας υποθέσουμε ότι το αρχείο με τα πιθανά username ονομάζεται usernames.txt, ενώ εκείνο

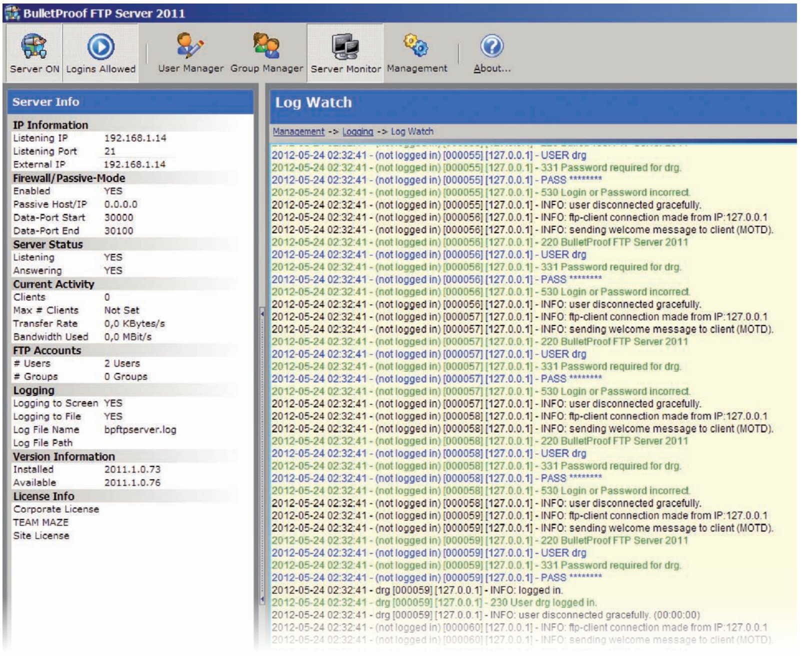
με τα πιθανά password είναι το passwords.txt. Για να δοκιμάσουμε επιτέλους το script, θα πρέπει να τοποθετήσουμε στην ίδια θέση (στο ίδιο path) τα δύο αρχεία κειμένου και να το ξεκινήσουμε κάπως έτσι:

```
python ftp_bruteforce.py localhost usernames.txt passwords.txt
```



Εικόνα 4
Τα αποτελέσματα της "ωμής επίθεσης" στον BulletProof FTP server. Τα passwords για τους χρήστες DeltaHacker και drg είναι SubZero και drg αντίστοιχα.

Πολύ σύντομα, το script θα πετύχει κάποιον από τους δύο σωστούς συνδυασμούς (drg:drg ή DeltaHakcer:SubZero) και θα μας ενημερώσει για την επιτυχία του. Τώρα, μπορούμε να ρίξουμε μια ματιά στα αρχεία καταγραφής του FTP server. Εκεί θα βρούμε ένα σύνολο απανωτών συνδέσεων, τις οποίες πραγματοποίησε ταχύτατα το σκριπτάκι. Βέβαια, αν τα usernames.txt και passwords.txt περιλαμβάνουν ελάχιστες λέξεις, οι συνδυασμοί που θα δοκιμάσει το πρόγραμμα θα είναι επίσης λίγοι. Επομένως μην περιμένετε να βρείτε πολλές απόπειρες σύνδεσης στα log του server. Σε ένα ρεαλιστικό σενάριο χρήσης, το passwords.txt θα ήταν σίγουρα πολύ μεγάλο και οι δοκιμαστικές συνδέσεις του script θα ήταν πάρα πολλές! Στην τρέχουσα μορφή του, το προγραμματάκι μας είναι πλήρως λειτουργικό. Ωστόσο αν θέλει κανείς να ασχοληθεί περισσότερο μαζί του, θα μπορούσε να το κάνει πιο αποδοτικό. Ένας τρόπος για να πετύχουμε κάτι τέτοιο είναι με πραγματοποίηση πολλών συνδέσεων



Εικόνα 5
Το log file αποδεικνύει εύκολα ότι κάποιος κακόβουλος προσπαθεί να εισβάλλει στον server. Αυτό το αρχείο εξετάζει ο administrator, για να γνωρίζει τι συμβαίνει στο μηχάνημά του και, αν το κρίνει σκόπιμο, να λάβει πρόσθετα μέτρα!

ταυτόχρονα. Αν έχετε και καμία άλλη ιδέα, περιμένουμε να την ακούσουμε στα fora του περιοδικού (http://deltahacker.gr/forums).

Επίθεση στο SSH

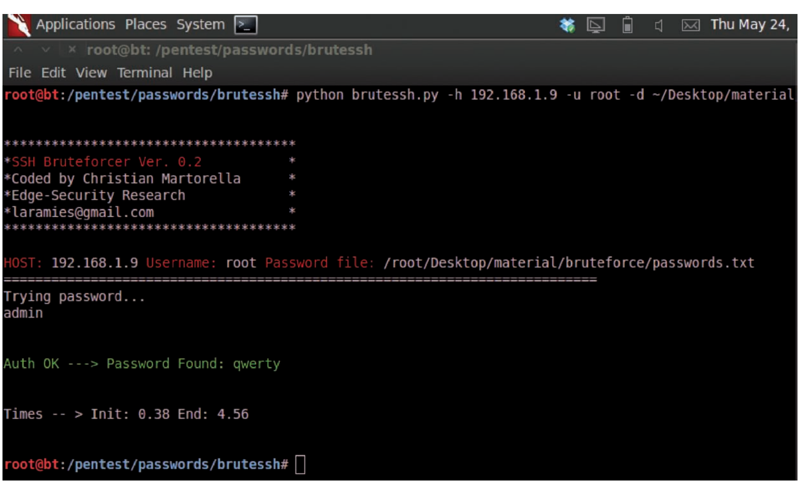
Ένα απλό εργαλείο για επιθέσεις bruteforce σε SSH servers, είναι το SSH-Bruteforce (περιμένατε πιο πρωτότυπο όνομα;). Μπείτε να το κατεβάσετε από εδώ: http://www.edge-security.com/soft/brutessh-0.5.tar.bz2. Εμείς το δοκιμάσαμε από ένα virtual machine με το Backtrack 5 R2 64bit και η επίθεση που κάναμε είχε στόχο έναν iMac. Αρχικά, λοιπόν, κατεβάσαμε το πακέτο με το script και το αποσυμπιέσαμε σε έναν φάκελο, στην επιφάνεια εργασίας. Το Backtrack VM είχε IP την 192.168.1.9 ενώ το iMac την 192.168.1.5. Έτσι, αφού μεταβήκαμε στον κατάλογο με το brutessh, δώσαμε το εξής:

python brutessh –h 192.168.1.5 –u root –d ~/Desktop/passwords.txt

Με την παράμετρο -d (από τη λέξη dictionary) δηλώνουμε το wordlist με τα πιθανά password. Στη δική μας περίπτωση ήταν το αρχείο passwords.txt, στη διαδρομή ~/Desktop. Η δε παράμετρος -h (host) χρησιμοποιείται για τη δήλωση του στόχου, ενώ με την παράμετρο -u δηλώνουμε το username που θα χρησιμοποιηθεί για τις επιθέσεις. Τα αποτελέσματα της επίθεσης απεικονίζονται στην εικόνα 7.

Hydra-gtk

Ένα ιδιαίτερα γνωστό εργαλείο στους κύκλους της ασφάλειας υπολογιστών είναι το hydra. Υποστηρίζει μια πληθώρα από πρωτόκολλα και είναι ιδιαίτερα ευέλικτο στις επιθέσεις bruteforce. Μας δίνει τη δυνατότητα να επιλέξουμε από έναν στόχο έως πολλαπλούς. Η GUI εκδοχή του είναι το Hydra-gtk. Όπως φαντάζεστε δεν λείπει από το Backtrack. Ακολουθήστε τη διαδρομή Privilege Escalation --> Password Attacks --> Online Attacks, επιλέξτε το Hydra-gtk και δοκιμάστε να “σπάσετε” τον FTP server που στήσαμε προηγουμένως. Δεν είναι και τόσο δύσκολο! Τις λεπτομέρειες τις αφήνουμε ως άσκηση -- για απορίες γράψτε στο forum του περιοδικού.



Εικόνα 6
Επιτυχημένη επίθεση με το εργαλείο brutessh, σε έναν iMac. Το password που είχε ο άτυχος root είναι qwerty. Τρελή φαντασία, ε;



Skill: Advanced

Tags: DNS, tunneling, iodine, hotspots

Πίσω από τις λέξεις, κρύβονται οι συνδέσεις!

by GiannouG

«DNS εκεί; Θέλω να μάθω τη διεύθυνση IP του τάδε site. Ή μάλλον, όχι! Μη μου δώσεις τη διεύθυνση. Πες μου για 'σένα. Θέλω να μιλήσω μαζί σου...» **Η αλήθεια είναι ότι δεν υπάρχει κανένας DNS, ο οποίος να κατανοεί αυτήν *ακριβώς* τη γλώσσα. Ωστόσο μπορούμε να στήσουμε έναν δικό μας name server, με τον οποίο θα επικοινωνούμε μέσω DNS queries. Θα μας φανεί πολύ πιο χρήσιμος απ' όσο φαντάζεστε. Πιστέψτε μας ;)**



Αχά! Αυτή η καφετέρια προσφέρει Internet μέσω hotspot. Κρίμα που θέλει χρήματα. Κρίμα γιατί δεν θα τα πάρει ;)

Έφτασε το Καλοκαίρι και μαζί του ήρθαν τα σχέδια για διακοπές. Λίγο-πολύ, όλοι θ' αφήσουμε τις βάσεις μας και θα εξορμήσουμε σε παραλίες, βουνά, χωριά κι όπου αλλού τη βρίσκει ο καθένας. (Σ.τ.Ε. Ρέικιαβικ!) Φυσικά, οι διακοπές είναι και καλές και αναγκαίες, αλλά προκαλούν μια κάποια αναστάτωση. Και μόνο η απομάκρυνση από το σπίτι μας, μπορεί μερικές φορές να μας ενοχλήσει. Δεν ξέρουμε τι παθαίνετε εσείς, δηλαδή, αλλά εμείς έχουμε πρόβλημα. Δεν μας ενοχλεί η απομάκρυνση από την (όποια) πόλη, φυσικά. Απλά, όταν περάσει κάτι από το μυαλό μας, υπάρχει ένας μόνο τρόπος για να ηρεμήσουμε: Να διερευνήσουμε αυτό το κάτι, να εξετάσουμε αν το έχουν σκεφτεί κι άλλοι, να μάθουμε αν το έχουν κάνει κι άλλοι και, γενικότερα, να μάθουμε τα πάντα γύρω από αυτό το κάτι. Όπως είναι προφανές, αυτή η διερεύνηση απαιτεί σύνδεση στο Internet αλλά αυτό δεν είναι καθόλου δεδομένο στους τόπους των διακοπών. Η όποια σύνδεση συνήθως παρέχεται από hotspots, σε συγκεκριμένα μέρη και με χρονοχρέωση. Η αλήθεια να λέγεται: Τις περισσότερες φορές οι τιμές είναι καλές για έναν έλεγχο στα mail και λίγο ψάξιμο/χάζεμα στο web. Τις περισσότερες φορές, αλλά όχι πάντα! Υπάρχουν και περιπτώσεις κατά τις οποίες η πρόσβαση στο Internet είναι μάλλον υπερτιμημένη. Τι γίνεται τότε; Είμαστε υποχρεωμένοι να πληρώσουμε τον κούκο αηδόνι, μόνο και μόνο για λίγη ώρα browsing;

Κάπου το πας, για συνέχισε...

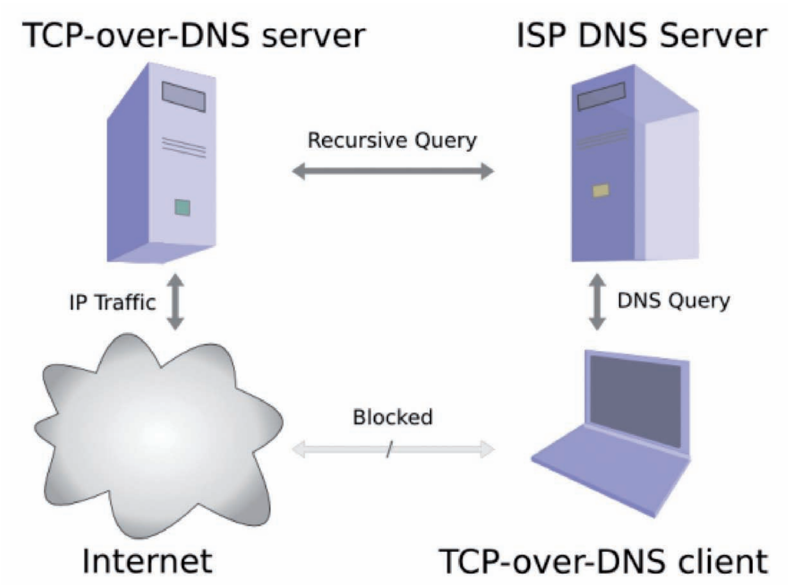
Τα hotspots είναι απλά μηχανήματα και μοιάζουν πολύ με τον router που έχουμε στο σπίτι μας: Μπορούμε να συνδεθούμε σε αυτά ασύρματα και να βγούμε στο Internet μέσω της δικής τους σύνδεσης. Βέβαια, αν δεν έχουμε πληρώσει το απαραίτητο αντίτιμο το hotspot παρεμποδίζει την επικοινωνία μας με τον έξω κόσμο (το Internet). Οποιαδήποτε προσπάθεια να επισκεφθούμε κάποιο site αποτυγχάνει και απλά μεταφερόμαστε σε μια ειδική σελίδα, όπου εκεί γίνεται η αγορά χρόνου πρόσβασης. Παρεμπιπτόντως, με την ίδια λογική λειτουργούν και τα "καρτοκινητά" πακέτα σύνδεσης στο Internet, των εταιριών κινητής τηλεφωνίας. Το μόνο που αλλάζει είναι το μέσο της επικοινωνίας. Αντί για WiFi υπάρχει μια σύνδεση 3G/HSDPA και αντί για τη σελίδα της εταιρίας που παρέχει το hotspot εμφανίζεται η σελίδα της εταιρείας κινητής τηλεφωνίας.

Η περαιτέρω περιγραφή της λειτουργίας ενός hotspot μοιάζει αδιάφορη, αλλά δεν είναι. Η συμπεριφορά τους κρύβει κάτι *πολύ* ενδιαφέρον. Η σελίδα της εταιρείας[1], εκεί δηλαδή όπου παρουσιάζονται οι χρεώσεις και ζητείται η αγορά χρόνου, δεν βρίσκεται στο τοπικό δίκτυο «πίσω» από το hotspot. Αντίθετα, αποτελεί τμήμα του site της εταιρείας και βρίσκεται στον έξω κόσμο --ναι, στο Internet! Πολλές φορές μάλιστα το hotspot επιτρέπει την πλήρη πρόσβαση στο site της εταιρίας, όπως και σε εκείνα διαφόρων χορηγών. Με λίγα λόγια, ακόμα κι όταν δεν έχουμε πληρώσει το hotspot επιτρέπει τη σύνδεσή μας στο Internet -- αλλά με περιορισμούς. Με περιορισμούς; Συγνώμη, αλλά εμείς δεν καταλαβαίνουμε και πολλά από περιορισμούς :D

DNS tunneling

Από ολόκληρο το Internet, το hotspot επιτρέπει την πρόσβαση σε συγκεκριμένες μόνο σελίδες. Μάλιστα οι διάφορες σελίδες εμφανίζονται με το "φυσιολογικό" τους URL. Δεν σερβίρονται από κάποιον τοπικό server, δηλαδή, ούτε βρίσκονται σε κάποιο "τοπικό" domain. Επιπρόσθετα, αν

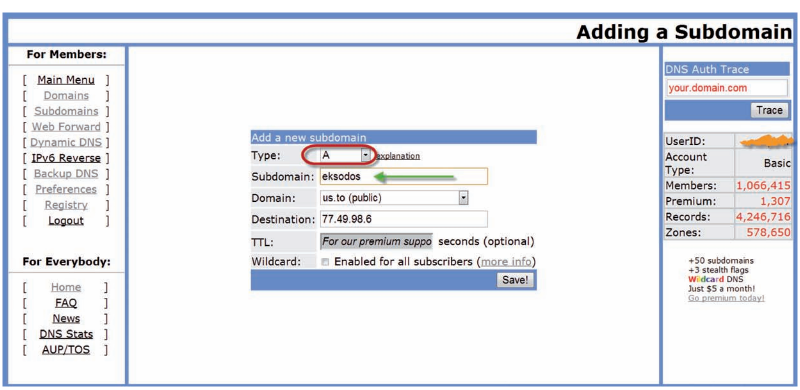
1. Του ξενοδοχείου, της καφετέριας ή οποιουδήποτε άλλου έχει τοποθετήσει το hotspot.



Η λογική του DNS tunneling είναι απλή. Οτιδήποτε θέλουμε να στείλουμε κωδικοποιείται και ενσωματώνεται σ' ένα DNS query, το οποίο εξασφαλίζουμε ότι θα φτάσει σε δικό μας name server. Αντίστοιχα, οτιδήποτε θέλει να μας στείλει ο server, κωδικοποιείται και τοποθετείται εντός των απαντήσεών του. Φυσικά, για να τα καταφέρουμε όλα αυτά χρειαζόμαστε έναν δικό μας name server, ο οποίος θα είναι υπεύθυνος για ένα ολόκληρο domain. Επομένως, θα πρέπει να έχουμε κι ένα δικό μας domain -- ή έστω subdomain.

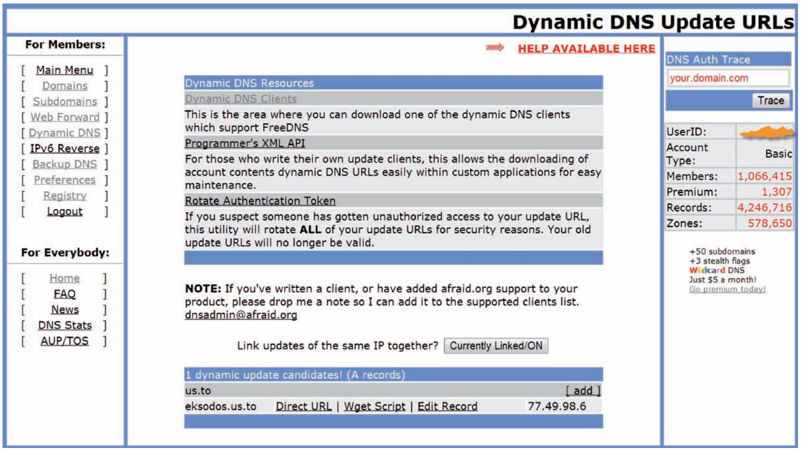
προσπαθήσουμε να συνδεθούμε σε κάποια άλλη σελίδα, θα λάβουμε ένα μήνυμα λάθους του στιλ: Η σύνδεση με το διακομιστή της ιστοσελίδας απέτυχε. Αντιληφθήκατε τη μικρή αλλά κρίσιμη λεπτομέρεια; Σε μια απόπειρα σύνδεσης, όταν δεν έχουμε πληρώσει αποτυγχάνει η σύνδεση με τον web server και όχι με τον name server. Με άλλα λόγια, τα ερωτήματα προς το DNS διακινούνται ελεύθερα!

Κάπου εδώ είναι ώρα να αφήσετε τη φαντασία σας ελεύθερη. Σκεφτείτε ότι κατέχετε ένα domain. Επίσης, υποθέστε ότι έχετε στήσει έναν name server, υπεύθυνο γι' αυτό το domain. Όταν βρίσκεστε πίσω από ένα hotspot, ανεξάρτητα από το αν έχετε πληρώσει ή όχι, μπορείτε να στείλετε ένα ερώτημα DNS, για τη διεύθυνση ενός μηχανήματος του domain σας. Αυτό το ερώτημα θα κινηθεί αρχικά προς τον DNS του ISP του hotspot κι από εκεί θα προωθηθεί στον DNS του domain σας. Με άλλα λόγια, μπορείτε πάντα να δημιουργείτε ερωτήματα DNS, τα οποία θα καταλήγουν στο δικό σας μηχάνημα (στον DNS του domain σας).



Αρχικά, πρέπει να δώσουμε ένα σταθερό όνομα στο μηχάνημα που θα φιλοξενεί τον «παράξενο» name server. Εμείς επιλέξαμε το όνομα eksodos, επειδή το συγκεκριμένο μηχάνημα θα λειτουργεί σαν έξοδος του DNS tunnel.

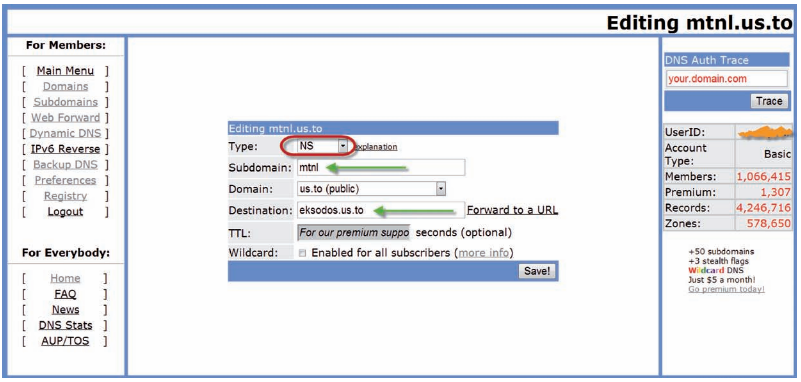
Αν το μηχάνημα με τον name server δεν έχει στατική IP, πρέπει να φροντίσουμε για την αυτόματη ενημέρωση του FreeDNS κάθε φορά που θα αλλάζει η διεύθυνσή του. Για το σκοπό αυτό, στο site της υπηρεσίας προσφέρονται πολλά scripts και για διάφορα λειτουργικά συστήματα: `http://freedns.afraid.org/scripts/freedns.clients.php`.



Συνεχίστε να φαντάζεστε. Σκεφτείτε έναν DNS, ο οποίος κατανοεί μια συνθηματική γλώσσα. Αυτή η συνθηματική γλώσσα είναι κατασκευασμένη με τέτοιο τρόπο, ώστε οι προτάσεις της να σχηματίζονται με τις ίδιες λέξεις που απαρτίζουν τα DNS queries και τις αντίστοιχες απαντήσεις των name server. Έτσι, μπορούμε να ανταλλάσουμε κωδικοποιημένα μηνύματα με τον name server, κρυμμένα μέσα σε DNS queries και στις αντίστοιχες απαντήσεις. Φοβερό, έτσι; Τύφλα να έχει ο James Bond, δηλαδή! Με έναν τέτοιο name server, θα μπορούσαμε άνετα να παρακάμπτουμε τους περιορισμούς ενός απλήρωτου hotspot...

Ας υποθέσουμε ότι έχετε ένα δικό σας domain κι έναν name server, με την παραπάνω εντυπωσιακή ικανότητα. Επιπρόσθετα, ας υποθέσουμε ότι αυτός ο name server αποτελεί τον κύριο DNS του domain σας. Με αυτό τον "εξοπλισμό" μπορείτε να ξεπεράσετε τους περιορισμούς όλων σχεδόν των Hotspot. Βλέπετε, τα πακέτα DNS διέρχονται ανενόχλητα από τα περισσότερα Hotspot. Επομένως, για να μιλήσετε με το server σας, αρκεί να δημιουργήσετε ερωτήματα DNS, τα οποία να αφορούν στο domain σας και εντός των οποίων να έχουν κωδικοποιηθεί τα μηνύματά σας! Εφόσον τα ερωτήματα έχουν κωδικοποιηθεί με τη γλώσσα που κατανοεί ο name server του domain σας, θα σας στείλει τα δικά του μηνύματα με τη μορφή απαντήσεων DNS. Έτσι, θα μπορείτε να μιλήσετε μαζί του ανεμπόδιστα.

Η παραπάνω τεχνική ονομάζεται DNS tunneling κι επιτρέπει να στέλνουμε και να λαμβάνουμε οποιουδήποτε είδους μηνύματα, μέσω πακέτων DNS. Πιο συγκεκριμένα, με το DNS tunneling μπορούμε να προωθούμε ολόκληρες συνδέσεις, ανεξαρτήτως πρωτοκόλλου, πάντα με τη βοήθεια ερωτημάτων DNS. Προϋπόθεση για να λειτουργήσει ένας τέτοιος μηχανισμός, είναι η ύπαρξη ενός ειδικού name server, ο οποίος θα ερμηνεύει κατάλληλα τα queries. Επίσης, εξίσου απαραίτητο είναι και το πρόγραμμα που θα κατασκευάζει τα κατάλληλα DNS queries, ενσωματώνοντας μέσα τους τις προς αποστολή πληροφορίες. Στη συνέχεια θα δούμε πώς μπορούμε να στήσουμε μόνοι μας ένα τέτοιο "σύστημα", ώστε να ξεπερνάμε τους περιορισμούς των ακριβών hotspot. Συγκεκριμένα, θα αναφερθούμε στο σετάρισμα ενός server με το Ubuntu 12.04 κι ενός φορητού με Windows 7. Το πρώτο μηχάνημα θα έχει το ρόλο του ειδικού name server, ο οποίος βγαίνει στο Internet ανενόχλητος (Σ.τ.Ε. VPS, ένιουαν;) ενώ το δεύτερο μηχάνημα είναι εκείνο από το οποίο παρακάμπτουμε τους περιορισμούς του εκάστοτε hotspot.



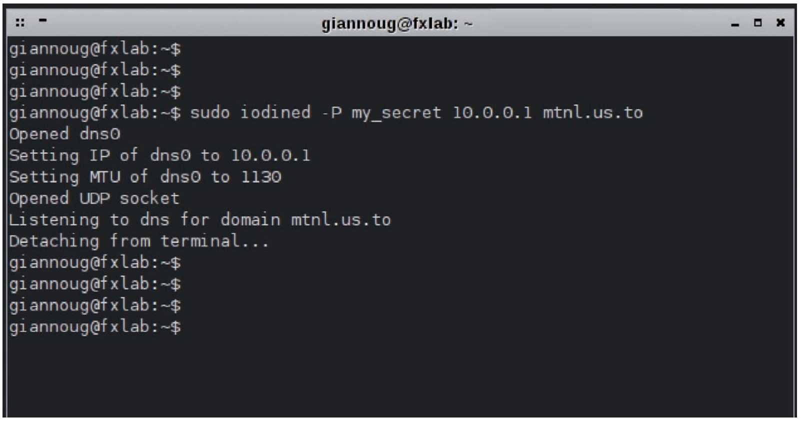
Εφόσον έχουμε δώσει ένα σταθερό όνομα στο μηχάνημα με τον name server, μπορούμε να δημιουργήσουμε ένα ακόμα subdomain. Σε αυτό θα δηλώσουμε ως κύριο name server τον δικό μας! Στην εικόνα φαίνεται η δημιουργία του subdomain mtnl.us.to, με κύριο name server το μηχάνημα eksodos.us.to.

Έξοδος από το τούνελ

Πριν φτιάξουμε το τούνελ και κυρίως πριν μπούμε μέσα, θα φροντίσουμε για την έξοδό του. Είναι που δεν θέλουμε να παγιδευτούμε ;) Όπως είπαμε, το μηχάνημα με την ελεύθερη πρόσβαση στο Internet --εκείνο που θα μας τροφοδοτεί με όσα του ζητάμε-- είναι ο name server του δικού μας domain. Αρχικά, λοιπόν, πρέπει να στήσουμε τη σχετική υπηρεσία σε κάποιον υπολογιστή μας. Όλη η δουλειά θα γίνεται από μια κατάλληλη εκδοχή name server, η οποία θα χειρίζεται τις συνδέσεις μας, θα αναγνωρίζει τα αιτήματά μας και θα απαντά κατάλληλα. Αναφερόμαστε στο περίφημο Iodine. Για την εγκατάστασή του στον μικρό μας server, αρκεί να δώσουμε κάτι τέτοιο:

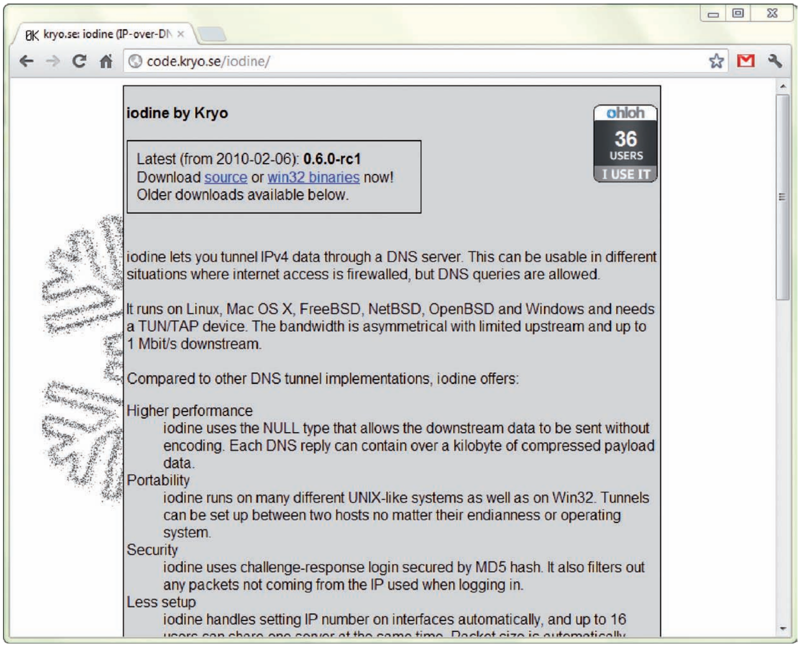
```
sudo apt-get update
sudo apt-get install iodine
```

Μόλις ολοκληρωθεί η εγκατάσταση, μπορούμε να προχωρήσουμε στη ρύθμιση και στην εκκίνηση της νέας μας υπηρεσίας. Αυτά τα δύο πραγματοποιούνται ταυτόχρονα σε μια κίνηση, αλλά πριν αναφερθούμε αναλυτικά πρέπει να ξεκαθαρίσουμε ορισμένα πράγματα. Όπως είπαμε νωρίτερα, για να δουλέψει το κολπάκι μας πρέπει να διαθέτουμε έναν name server, ο οποίος να είναι υπεύθυνος για ένα ολόκληρο domain. Βέβαια, η απόκτηση ενός ολόκληρου domain δεν γίνεται δωρεάν και η πλήρης διαχείρισή του δεν είναι εύκολη. Ωστόσο μπορούμε να αποκτήσουμε ένα subdomain, σε κάποιο από τα λεγόμενα public domains (τα ελεύθερα



Στον server θα πρέπει να εγκαταστήσουμε το iodine και να ξεκινήσουμε το σχετικό daemon (iodined). Για την εκκίνησή του πρέπει να δώσουμε το όνομα του subdomain και τη διεύθυνση IP, η οποία θα αντιστοιχεί στο server εντός του τούνελ. Προαιρετικά, μπορούμε να προστατεύσουμε το τούνελ με ένα συνθηματικό.

*Όταν ξεμπερδέψουμε με το server, πρέπει να εγκαταστήσουμε το iodine και στο μηχάνημα από το οποίο θα δημιουργούμε τα DNS tunnels. Ευτυχώς, το iodine προσφέρεται *και* για Windows ;)*



domain), εντελώς δωρεάν. Αυτό το subdomain, μάλιστα, θα μπορούμε να το ρυθμίσουμε πανεύκολα και να του ορίσουμε σαν κύριο name server το μηχάνημα στο οποίο εγκαταστήσαμε προηγουμένως το iodine! Αναρωτιέστε πώς μπορούμε να πετύχουμε όλα αυτά; Μια καλή λύση αποτελεί το freeDNS (`http://freedns.afraid.org`). Το επόμενο βήμα, λοιπόν, είναι η δημιουργία ενός λογαριασμού (user account) στην εν λόγω υπηρεσία...

Ταυτοποιήσεις

Σύμφωνα με τους δημιουργούς του iodine, όσο πιο μικρό είναι το domain name, του οποίου την ευθύνη έχει ο name server μας, τόσο πιο αποδοτικά λειτουργεί το σύστημα. Έτσι, αφού συνδεθούμε στο freeDNS, πρέπει να επισκεφθούμε την περιοχή Registry. Εκεί εμφανίζονται όλα τα ελεύθερα domain και οφείλουμε να επιλέξουμε εκείνο με το συντομότερο όνομα. Εμείς επιλέξαμε το "us.to". Με ένα κλικ επάνω στο domain, μεταφερόμαστε σε μια ειδική περιοχή του site από την οποία μπορούμε να δημιουργήσουμε αυθαίρετα ένα δικό μας subdomain, εντός του επιλεγμένου domain. Αρχικά, πρέπει να εξασφαλίσουμε ότι το μηχάνημα στο οποίο τρέχει ο iodine server, θα αποκτήσει ένα σταθερό όνομα. Γι' αυτό το λόγο, πρέπει να δημιουργήσουμε ένα subdomain, το οποίο θα παραπέμπει στον server μας. Εμείς δημιουργήσαμε ένα subdomain με το όνομα "eksodos" και ορίσαμε σαν destination τη διεύθυνση IP του μηχανήματός μας. Αφού αποθηκεύσαμε τις αλλαγές και περιμέναμε λίγη ώρα (για να ενημερωθούν οι DNS του κόσμου όλου), το μηχάνημά μας ήταν πλέον προσβάσιμο με το όνομα eksodos.us.to. Παρεμπιπτόντως, αν ο server μας βρίσκεται πίσω από κάποιο router και δεν είναι --λέμε τώρα-- ένα ωραιότατο VPS, θα πρέπει να φροντίσουμε τις ακόλουθες δύο λεπτομέρειες:

Επειδή η διεύθυνση που έχουμε από τον ISP αλλάζει κατά διαστήματα, πρέπει να εξασφαλίσουμε ότι το freeDNS θα ενημερώνεται για την όποια

αλλαγή. Για το σκοπό αυτό, στην περιοχή "Dynamic DNS clients" του FreeNDS (http://freedns.afraid.org/scripts/freedns.clients.php) παρέχονται πολλά script, τα οποία αυτοματοποιούν τη σχετική διαδικασία.

Ο server μας βρίσκεται πίσω από router, αλλά πρέπει να εξυπηρετεί συνδέσεις που έρχονται από το Internet. Τι πρέπει να κάνουμε, λοιπόν; Μα, port forwarding, φυσικά! Συγκεκριμένα, εφόσον ο server μας είναι DNS, θα πρέπει να προωθήσουμε το port 53 του router προς το μηχάνημά με τον iodine server.

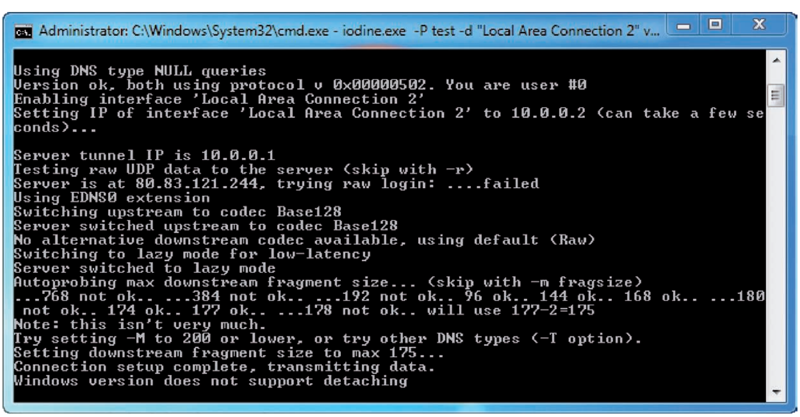
Εφόσον έχουμε δώσει ένα σταθερό όνομα στον name server, μπορούμε να προχωρήσουμε στη δημιουργία ενός ακόμα subdomain[2]. Αυτό το subdomain θα πρέπει να έχει σύντομο όνομα, όπως και το domain. Πρόκειται για το subdomain την ευθύνη του οποίου θα αναλάβει ο name server μας. Εδώ χρειάζεται λίγη προσοχή: Κατά τη δημιουργία αυτού του subdomain (στο freeDNS), πρέπει να επιλέξουμε ως τύπο το "NS", και στο πεδίο destination να δώσουμε το όνομα του name server μας. Εμείς δημιουργήσαμε το subdomain mtnl (από το My TuNneL) και ορίσαμε σαν destination το eksodos.us.to. Με αυτόν τον τρόπο, δημιουργήθηκε το subdomain mtnl.us.to, με κύριο name server το eksodos.us.to. Μεγάλη σπαζοκεφαλιά, έτσι;

Ολοκλήρωση του server

Τέλεια! Ο name server μας είναι σχεδόν έτοιμος. Το μηχάνημα που τον φιλοξενεί έχει ένα σταθερό και γνωστό σε όλο το Internet όνομα (eksodos.us.to), ενώ έχει επιλεγεί σαν κύριος name server για το subdomain mtnl.us.to. Τώρα, μπορούμε να ξεκινήσουμε το iodine και να ξεμπερδέψουμε (επιτέλους) με τον DNS. Σας είπαμε πόσο απλό είναι το iodine, έτσι; Για τη δημιουργία ενός tunnel χρειάζεται να δώσουμε ελάχιστα στοιχεία. Δείτε πώς ξεκινήσαμε το iodine εμείς:

```
iodined -P my_secret 10.0.0.1 mtnl.us.to
```

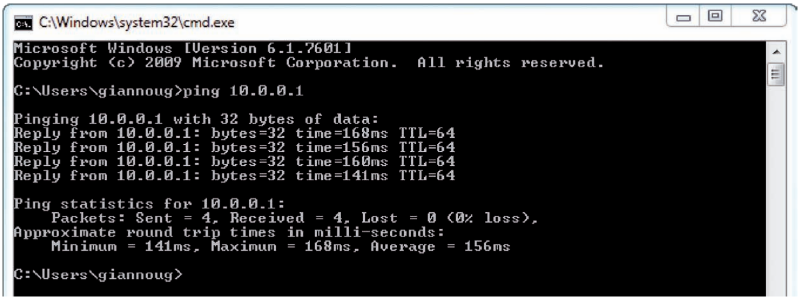
Καταρχάς, παρατηρείστε το τελικό d στο iodined. Δεν ξεκινήσαμε την εφαρμογή client για σύνδεση σε DNS tunnel, αλλά το iodine daemon, για τη δημιουργία DNS tunnel. Η πρώτη παράμετρος που δώσαμε αποτελεί έναν κωδικό, για την είσοδο στο τούνελ. Όπως αντιλαμβάνεστε, δεν υπάρχει κανένας λόγος να ορίσουμε δύσκολο συνθηματικό. Στη συνέχεια



Στο μηχάνημα από το οποίο θα δημιουργούμε τα τούνελ, εκτός από το iodine θα χρειαστούμε και το OpenVPN GUI for Windows. Αφού εγκατασταθεί κι αυτό, μπορούμε να ξεκινήσουμε το iodine. Προς τούτο, εκτελούμε το ομώνυμο πρόγραμμα (χωρίς τελικό d), με παράμετρο τη συσκευή TAP/TUN του OpenVPN GUI, το subdomain στο οποίο δραστηριοποιείται ο name server μας και το αντίστοιχο συνθηματικό.

2. Στο FreeDNS, πάντα, κάτι τέτοιο γίνεται πανεύκολα από την περιοχή... Subdomains. Εντελώς απρόσμενο, ε;

Για να ελέγξουμε αν το τούνελ δημιουργήθηκε και λειτουργεί σωστά, μπορούμε να κάνουμε στον server μας ένα ping. Εφόσον έχει δημιουργηθεί το τούνελ, ο server θα έχει αποκτήσει την IP που ορίσαμε κατά την εκκίνηση του iodine server. Έτσι, η δοκιμή μας θα πρέπει να γίνει με τη σχετική διεύθυνση IP (στην περίπτωσή μας ήταν η 10.0.0.1).



ορίζουμε τη διεύθυνση IP, στην οποία θα ακούει ο server μας *εντός* του tunnel. Αυτή τη διεύθυνση την επιλέγουμε αυθαίρετα, αλλά πρέπει να ανήκει σε κάποιο από τα IP ranges τα οποία χρησιμοποιούνται στα τοπικά δίκτυα (πρέπει να είναι private IP, με άλλα λόγια). Τέλος, δηλώνουμε το subdomain την ευθύνη του οποίου θα έχει ο name server μας. Αυτό ήταν όλο! Πλέον, μπορούμε να στραφούμε στο λαπτοπάκι με τα Windows 7, από το οποίο θα βγαίνουμε στο Internet όταν βρισκόμαστε πίσω από ένα hotspot.

Είσοδος στο τούνελ

Για τη δημιουργία του τούνελ χρειαζόμαστε ένα tunnel device (TUN/TAP device), όπως και στην περίπτωση του OpenVPN (βλ. και σχετικό άρθρο στο deltaHacker 007). Ένας τρόπος για να το πετύχουμε είναι να εγκαταστήσουμε το OpenVPN GUI for Windows. Καλό θα ήταν μάλιστα να πάρουμε την πιο πρόσφατη έκδοσή του: http://openvpn.se/download.html. Στο τέλος της εγκατάστασης, δημιουργείται αυτόματα μια εικονική συσκευή δικτύου. Στις δικές μας δοκιμές, αυτή η συσκευή πήρε το όνομα Local Area Connection 2. Τώρα, μπορούμε να κατεβάσουμε το iodine για Windows από εδώ: http://code.kryo.se/iodine. Αφού το εγκαταστήσουμε, πρέπει να ανοίξουμε ένα τερματικό (cmd.exe) με δικαιώματα administrator και να μεταβούμε στον κατάλογο bin που βρίσκεται μέσα στον κατάλογο του iodine. Από εκεί θα πρέπει να τρέξουμε το iodine (χωρίς τελικό d). Η ρύθμιση του iodine για τη δημιουργία του DNS tunnel είναι πολύ απλή. Δείτε πώς το εκκινήσαμε εμείς:

```
iodine.exe -P my_secret -d «Local Area Connection 2» \ mtnl.
us.to
```

Όπως αντιλαμβάνεστε, η πρώτη παράμετρος είναι το συνθηματικό που ορίσαμε για τον server, ενώ η δεύτερη παράμετρος ορίζει το όνομα της συσκευής TUN/TAP. Τέλος, η τρίτη παράμετρος δηλώνει το subdomain, του οποίου κύριος name server έχει οριστεί ο δικός μας! Μόλις ξεκινήσει το πρόγραμμα, ο υπολογιστής μας θα μιλήσει με τον DNS του ISP του hotspot και, στη συνέχεια, με τον DNS του subdomain μας (τον iodine server). Έτσι, μέσα σε λίγα δευτερόλεπτα θα δημιουργηθεί το tunnel. Τώρα, για να ελέγξουμε ότι λειτουργούν όλα σωστά, μπορούμε να κάνουμε ένα ping στον server μας:

```
ping 10.0.0.1
```

Για τη δοκιμή χρησιμοποιούμε τη διεύθυνση IP που επιλέξαμε κατά την εκκίνηση του iodine server. Αν πετύχει το ping, τότε το τούνελ έχει δημιουργηθεί σωστά και το μηχάνημά μας βρίσκεται εντός του.

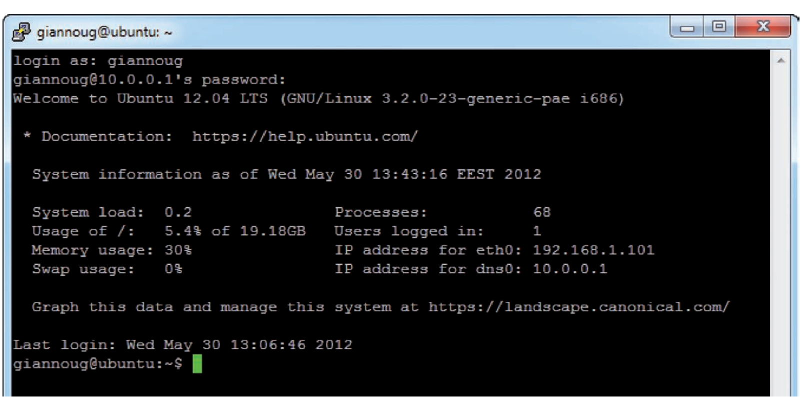
Απόδραση από το hotspot

Για την ώρα έχουμε καταφέρει να συνδεθούμε με τον server μας. Βλέπετε, εκτός από ping μπορούμε εξίσου εύκολα να συνδεθούμε και με SSH. Τι γίνεται όμως με το υπόλοιπο Internet; Πώς μπορούμε να επισκεφθούμε και άλλους τόπους; Η πρώτη και πιο απλή μέθοδος, με μέτρια αποτελέσματα, είναι η χρήση κάποιου SSH tunnel, μέσα στο ήδη υπάρχον DNS tunnel! Αν θέλετε να μάθετε περισσότερα για τα SSH tunnels και πώς να τα χρησιμοποιείτε, μπορείτε να διαβάσετε το σχετικό άρθρο στο Parabing: http://parabing.com/?p=552. Μία ακόμα λύση είναι η χρήση κάποιου proxy. Θα μπορούσε κάποιος να σηκώσει έναν web server (είτε Apache, είτε lighttpd) στον server που φιλοξενεί το iodined και, μέσω αυτού, να επισκέπτεται όλες τις σελίδες που θέλει. Όμως, για να δουλέψει αυτό το τρικ απαιτείται κι ένα PHP proxy script! Όπως καταλαβαίνετε, πρόκειται για περίπλοκη λύση. Μία καλύτερη λύση θα αποτελούσε η χρήση κάποιου software proxy, όπως το Squid. Ωστόσο, οι απαιτήσεις του σε μνήμη καθιστούν δύσκολη τη χρήση του, σε έναν μικρό VPS (Σ.τ.Ε. Υπάρχουν και φθηνά VPSes με 512MB RAM. Δείτε, π.χ., τις σχετικές προτάσεις στο http://bsdvm.com.) Τέλος, μια άλλη λύση θα ήταν να δημιουργήσουμε ένα δίκτυο VPN μέσα από το DNS tunnel! Αν μπορείτε να φανταστείτε κι άλλες λύσεις, περιμένουμε να τις ακούσουμε.

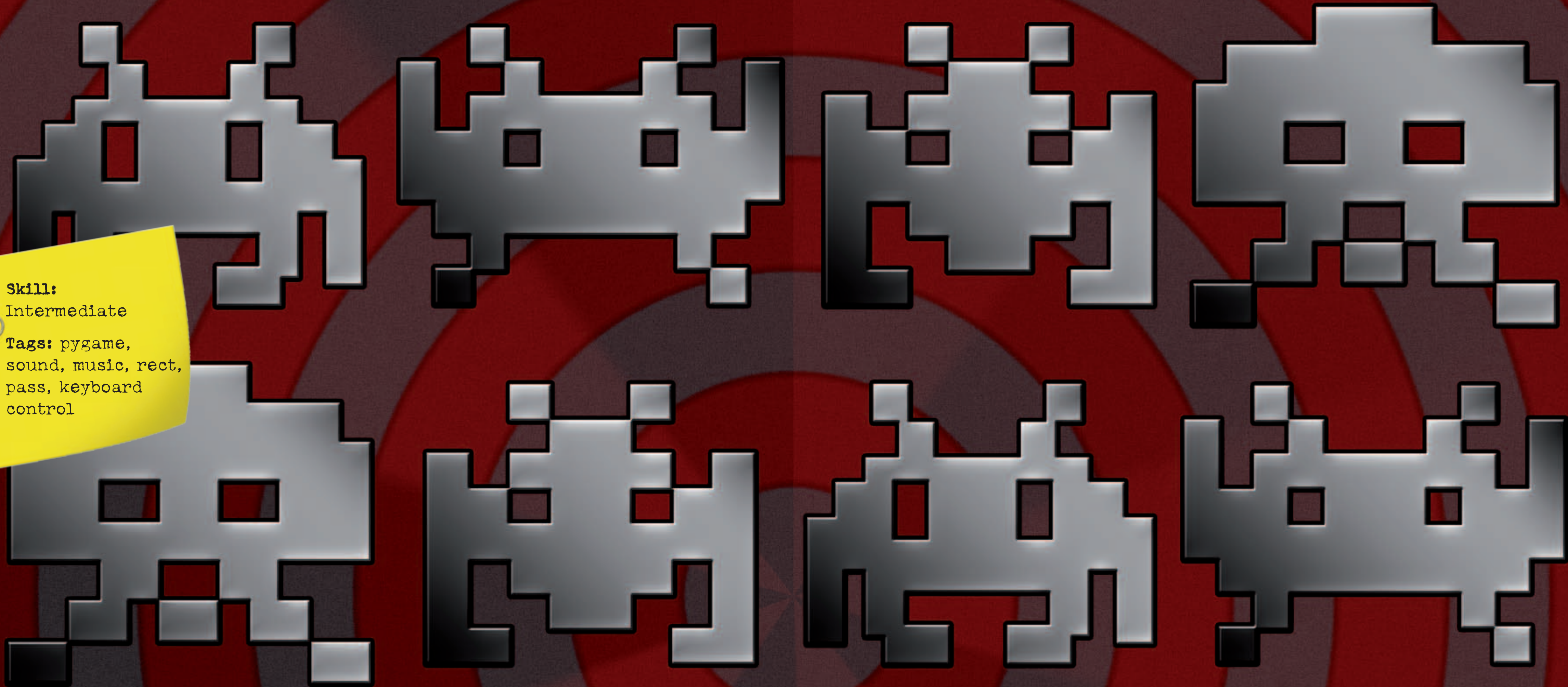
Τελικές σκέψεις...

Υπάρχουν περιπτώσεις, για παράδειγμα το δίκτυο μιας γνωστής εταιρίας κινητής τηλεφωνία, στις οποίες οι τεχνικοί έχουν φροντίσει να κλείσουν την «τρύπα» του DNS. Μετά από δοκιμές που κάναμε, φάνηκε πως τα πακέτα DNS δεν απορρίπτονται αλλά... περιορίζονται. Έτσι, το tunnel δημιουργείται κανονικά αλλά μετά από λίγο η ταχύτητά του περιορίζεται δραματικά! Εμείς πετύχαμε ταχύτητες που θύμιζαν παλιά, αναλογική σύνδεση των 56Kbps. Αρκετά καλή για τον έλεγχο των email αλλά τοξική για την ψυχική υγεία, όταν περνάει δεκάλεπτο. Γενικότερα, το DNS tunneling αποτελεί μια παρατραβηγμένη, αλλά μάλλον πετυχημένη λύση για την παράκαμψη συγκεκριμένων περιορισμών. Βλέπετε, η υπηρεσία DNS είναι απαραίτητη σε κάθε δίκτυο, οπότε τις περισσότερες φορές παραμένει ξεκλείδωτη.

Σας αφήνουμε να φτιάξετε και να παίξετε με τα δικά σας DNS tunnels, πίνοντας μια γευστικότατη πορτοκαλάδα, με τα λεφτά που γλυτώσατε από το Internet!



Εφόσον έχει στηθεί το τούνελ μπορούμε να συνδεθούμε στον server μας και με άλλα πρωτόκολλα. Για παράδειγμα, μπορούμε να συνδεθούμε και μέσω SSH.



Skill: Intermediate

Tags: pygame, sound, music, rect, pass, keyboard control

Python Game Programming:

Pygame Invaders!



by Sonic

Αν είχατε ξεκινήσει να προγραμματίζετε στα 80s, είναι σίγουρο ότι τα πρώτα σας προγράμματα θα ήταν παρόμοια με αυτά που έχουμε δει μέχρι τώρα. Πολλά από αυτά θα τα είχατε αντιγράψει από το εγχειρίδιο προγραμματισμού του υπολογιστή σας και σίγουρα θα είχατε σπαταλήσει αρκετές ώρες για να κατανοήσετε πώς λειτουργούν και να πειραματιστείτε. Μάλιστα, αν μας επιτρέπετε να μαντέψουμε, μετά από τα πρώτα σας προγράμματα (των 5 ως 10 γραμμών), θα είχατε γράψει ένα "Guess the Number", ένα "Bouncing Ball" κι ένα "Graphics Match".

Δεν ξέρω αν η πορεία αυτή σάς θυμίζει κάτι. Όπως και να ‛χει, σε αυτό το άρθρο της σειράς μας θα προχωρήσουμε ένα βήμα παραπέρα. Ο γράφων ξεκίνησε με απλά προγράμματα, σαν αυτά που έχουμε ήδη δει, και συνέχισε με τις δικές του παραλλαγές οι οποίες γρήγορα κατέληξαν σε νέα, πρωτότυπα προγράμματα. Φυσικά, μια άλλη πηγή έμπνευσης ήταν τα περιοδικά (Pixel, Micromad κ.ά.), που δημοσίευαν listings προγραμμάτων. Οι νέοι προγραμματιστές της εποχής πληκτρολογούσαν αυτά τα listings κι αντλούσαν ατελείωτες ιδέες αλλά και λύσεις στα δικά τους προγραμματιστικά προβλήματα (Παράδειγμα: Πώς θα κάνω το καταραμένο το διαστημοπλοιάκι να κινείται με τα βελάκια;)

Αυτά που θέλω να τονίσω με τα παραπάνω είναι:

- Κανείς δεν γεννήθηκε γνωρίζοντας προγραμματισμό. Όλοι μάθαμε διαβάζοντας, πειραματιζόμενοι και παίζοντας με τον κώδικα άλλων. Αυτό που μας «χωρίζει» με όσους δεν έμαθαν προγραμματισμό ή έστω δεν έχουν μάθει ακόμα, είναι η θέληση και η προσπάθεια να τα καταφέρουμε.
- Κατά τη συγγραφή ενός προγράμματος, είναι πολύ συνηθισμένο να αλλάζουμε τον κώδικα σε τέτοιο βαθμό, ώστε να μην έχει καμία σχέση με τον αρχικό! Αυτό θα το διαπιστώσετε και μόνοι σας, καθώς θα φτιάχνουμε το τελικό μας παιχνίδι. Παρεμπιπτόντως, αυτό το παιχνίδι ξεκινάει *τώρα*!

Έχουμε μάθει λίγο προγραμματισμό και αλγοριθμική, λίγο από object oriented design και φυσικά τεχνικές που χρειάζονται στα παιχνίδια, όπως η ανάγνωση πληκτρολογίου. Έχουμε πλέον τη γνώση και το θάρρος (!) να δοκιμάσουμε να γράψουμε το δικό μας. Τι παιχνίδι, όμως; Όπως είπα νωρίτερα, τις παλιές καλές εποχές παίρναμε ιδέες από προγράμματα άλλων, όπως και από τα παιχνίδια που βλέπαμε στις αίθουσες με τα ηλεκτρονικά (ουφάδικα, όπως τα λέγαμε τότε). Τα πιο πολλά σχετίζονταν με κάποια εισβολή εξωγήινων. Ήταν, δηλαδή, τα κλασικά shoot ‛em up. Ε, λοιπόν,



Εικόνα 1
Ο μαθητής Ανδρέας Κοντορίνης παρουσιάζει το Pygame, στην Έκθεση Μαθητικής Δημιουργίας Πληροφορικής στα Χανιά. Ο Ανδρέας υπήρξε βασικό πειραματ, εχμ, beta tester, για τη σειρά μαθημάτων Python Game Programming.

τριάντα χρόνια μετά (και βάλε), οι εξωγήινοι δεν ήρθαν ποτέ. Στην πραγματικότητα, αποκαλύφθηκε ότι τους γήινους πρέπει να φοβόμαστε περισσότερο. Για εμάς όμως ο στόχος παραμένει: Το πρώτο μας πραγματικό παιχνίδι πρέπει να είναι shoot 'em up και μάλιστα με τον τίτλο «Pygame Invaders!"

Πόσο δύσκολο είναι;

Πριν αρχίσουμε να σκεφτόμαστε τη δυσκολία και τρομοκρατηθούμε, πρέπει να περιγράψουμε το παιχνίδι και να ξεκαθαρίσουμε όλα τα χαρακτηριστικά που θέλουμε να έχει:

- Θα παίζεται σ' ένα παράθυρο 480x640 (portrait)
- Το (υπερσύγχρονο) διαστημοπλοιάκι θα κινείται αριστερά ή δεξιά, στο κάτω μέρος της οθόνης, και θα πυροβολεί ταχύτατα.
- Οι εξωγήινοι θα εμφανίζονται στην οθόνη κατά κύματα, θα ρίχνουν αδιακρίτως κι ενώ δεν θα είναι ιδιαίτερα έξυπνοι, θα υπερτερούν αριθμητικά.
- Το διαστημόπλοιο μας θα διαθέτει ασπίδα, η οποία θα μειώνεται με κάθε βολή που δέχεται. Το παιχνίδι θα τελειώνει (θα χάνουμε), όταν η ασπίδα πέφτει στο μηδέν. Ωστόσο, ανά 100 βαθμούς, θα κερδίζουμε μια μονάδα ασπίδας. Αντίθετα, οι εξωγήινοι θα πεθαίνουν με μια βολή (ούτε έξυπνοι, ούτε ανθεκτικοί: δικό μας είναι το πρόγραμμα, ό,τι θέλουμε τους κάνουμε). Για καθέναν που θα ξεπαστρεύουμε, θα παίρνουμε 10 βαθμούς!
- Το φόντο είναι μαύρο, με αστέρια. (Τι περιμένετε δηλαδή; Στο διάστημα είναι το παιχνίδι!) Τα αστέρια θα θέλαμε να κινούνται (background scrolling).
- Το παιχνίδι θα διαθέτει ηχητικά εφέ και μουσική.

Αν ρωτήσετε πόσο πιο εύκολο είναι να γραφτεί αυτό το πρόγραμμα σήμερα σε σχέση με το 80, θα σας απαντήσω το εξής: Αρκετά πιο εύκολο! Με τον επεξεργαστή των 3Mhz του 1981 και χωρίς δυνατότητα δημιουργίας καρέ, το scrolling background γίνεται ένα απατηλό όνειρο -- εκτός αν πάμε σε Assembly. Όσο για τη γλώσσα προγραμματισμού, η BASIC της εποχής δεν διέθετε objects. Η απλότητα της γλώσσας σήμαινε ότι δεν χρειαζόταν να μάθουμε το χειρισμό πολύπλοκων εντολών, συναρτήσεων και άλλων δομών. Από την άλλη, όμως, για να πετυχαίνουμε τους στόχους μας έπρεπε συχνά να καταφεύγουμε σε περίεργες κι ανορθόδοξες τεχνικές. Αντίθετα, στο pygame μπορούμε να γράφουμε τα πάντα κι ο υπολογιστής μας αντεπεξέρχεται στα σίγουρα. Επίσης, το παιχνίδι θα γραφτεί με τη λογική του object oriented programming και ο κώδικας θα είναι ξεκάθαρος και δομημένος.

Ήχος και Pygame

Το παιχνίδι μας θέλουμε να έχει ηχητικά εφέ και μουσική. Αυτό είναι κάτι που δεν έχουμε εξετάσει στο pygame ακόμα -- και μάλλον έφτασε η ώρα. Υπάρχουν δύο βασικοί τρόποι για την αναπαραγωγή ήχου.

- Το **pygame.mixer.music** χρησιμοποιείται συνήθως για τη μουσική που θέλουμε να ακούγεται στο background. Το αρχείο ήχου που θα δηλώσουμε δεν φορτώνεται ολόκληρο στη μνήμη, αλλά σταδιακά (κάτι σαν τοπικό streaming).

- Το **pygame.mixer.Sound** χρησιμοποιείται για ηχητικά εφέ. Μας επιστρέφει ένα αντικείμενο τύπου «sound» (τι παράξενο!) το οποίο, ανάμεσα σε άλλες, διαθέτει τη μέθοδο play. Μπορούμε να ετοιμάσουμε όσα αντικείμενα sound χρειαζόμαστε και να αντιστοιχήσουμε στο καθένα ένα διαφορετικό ηχητικό εφέ. Αργότερα, θα μπορούμε να καλούμε τη μέθοδο play του καθενός, στο κατάλληλο σημείο του παιχνιδιού.

Επειδή λίγες γραμμές κώδικα αξίζουν όσο εκατό περιγραφές (Σ.τ.Ε. Να υποθέσουμε ότι κάθε περιγραφή έχει 10 λέξεις; #troll), δείτε τα παρακάτω παραδείγματα και θα μπείτε αμέσως στο νόημα:

```
# Sound effects
import pygame
from pygame.locals import *
pygame.init()
clock = pygame.time.Clock()
laser = pygame.mixer.Sound(«laser.wav»)
laser.play()
while pygame.mixer.get_busy():
    clock.tick()
```

Στο παραπάνω απόσπασμα δημιουργούμε ένα αντικείμενο sound με το όνομα laser. Σε αυτό το αντικείμενο αντιστοιχούμε το αρχείο «laser.wav”, το οποίο πρέπει να βρίσκεται στον τρέχοντα κατάλογο. Μπορείτε να το κατεβάσετε μαζί με όλα τα άλλα αρχεία του άρθρου από το http://bit.ly/dh009invaders. Στη συνέχεια καλούμε τη μέθοδο play και ξεκινάμε ένα βρόχο while, ο οποίος θα ολοκληρωθεί όταν η pygame.mixer.get_busy() επιστρέψει false (δηλαδή όταν ολοκληρωθεί η αναπαραγωγή του «laser.wav”). Ο βρόχος εισάγει μια καθυστέρηση πριν τερματιστεί το πρόγραμμα, μόνο και μόνο για να ακουστεί ο ήχος! Στο κανονικό παιχνίδι δεν θα χρειαστούμε κάτι τέτοιο, εφόσον θα εκτελείται ο βασικός βρόχος του παιχνιδιού (το πρόγραμμα δεν θα τελειώνει αμέσως μετά την εκκίνηση της αναπαραγωγής). Ας δούμε ένα ακόμα παράδειγμα:

```
# Background music
import pygame
from pygame.locals import *
pygame.init()
clock = pygame.time.Clock()
pygame.mixer.music.load(«spaceinvaders.ogg»)
pygame.mixer.music.play()
while pygame.mixer.music.get_busy():
    clock.tick()
```

Εδώ δεν δημιουργούμε κάποιο νέο/πρόσθετο αντικείμενο. Βλέπετε, το pygame παρέχει μόνο ένα κανάλι για το streaming μουσικής. Έτσι, υποδεικνύουμε το αρχείο μουσικής που θέλουμε να παίζει ο mixer του pygame και απλά ξεκινάμε την αναπαραγωγή του. Είναι πολύ ενδιαφέρον το ότι μπορούμε να δώσουμε κάτι τέτοιο:

```
pygame.mixer.music.play(6)
```



Εικόνα 2
Το περίπτερό μας στην έκθεση. Ο original TI-99/4A με το cartridge TI Invaders αλλά και με παιχνίδια του γράφοντα σε BASIC κι Extended BASI,C διασκέδασε πολύ κόσμο. Τελικά, το κλασικό δεν πεθαίνει ποτέ. Εξάλλου, κατά πώς φαίνεται το συγκεκριμένο μηχάνημα είναι φτιαγμένο για να αντέξει για πάντα. Στα δεξιά, ο βίντεο-προβολέας δείχνει το hit της έκθεσης: Το Pygame Invaders!

Σε αυτή την περίπτωση το τραγούδι θα επαναληφθεί έξι φορές. Αν θέλουμε να επαναλαμβάνεται συνεχώς, πρέπει να δώσουμε αυτό:

```
pygame.mixer.music.play(-1)
```

Τέλος, να σημειώσουμε ότι το pygame δεν τα πάει πολύ καλά με τα MP3. Καλό θα ήταν να προτιμήσετε αρχεία OGG για τη μουσική και ασυμπίεστα WAV για τα πολύ πιο σύντομα ηχητικά εφέ.

Τα αντικείμενα του παιχνιδιού

Έχοντας ξεκαθαρίσει κάπως τα πράγματα με τον ήχο, μπορούμε να περάσουμε σε πιο σοβαρά ζητήματα: Ποια αντικείμενα και ποιες κλάσεις θα χρειαστεί να δημιουργήσουμε; Ας καταγράψουμε τα κινούμενα γραφικά:

- Το διαστημοπλοιάκι
- Οι εξωγήινοι
- Το δικό μας laser
- Το laser των εξωγήινων (μακριά από μας!)
- Το φόντο

Ίσως να μπείτε στον πειρασμό να φτιάξετε μια κλάση για το καθένα από αυτά, αλλά μη βιαστείτε. Όχι ότι θα είναι λάθος αν το κάνετε, αλλά θα βρεθείτε να γράφετε τον ίδιο κώδικα δύο φορές. Αν το σκεφτείτε καλύτερα, θα διαπιστώσετε ότι το διαστημοπλοιάκι μας και οι εξωγήινοι είναι παρόμοια αντικείμενα και χρειάζονται παρόμοιες μεθόδους. Επίσης, οι δικές μας βολές laser διαφέρουν ελάχιστα από εκείνες των εξωγήινων: Έχουν μόνο διαφορετική κατεύθυνση (και στόχο).

Ποιες μεθόδους θα χρειαστούμε για το δικό μας σκάφος; Θυμηθείτε την object oriented εκδοχή του bouncing ball, στο μέρος 4.5 της σειράς (`http://deltahacker.gr/?p=5495`). Για τα διαστημόπλοια, γενικά, θα χρειαστούμε τις ακόλουθες μεθόδους.

- **Show:** Για την απεικόνιση σε μια επιφάνεια
- **Move:** Για την κίνηση
- **Fire:** Για την βολή (ΟΚ, αυτό είναι καινούριο! Δεν έχουμε δει πουθενά bouncing balls που να πυροβολούν)
- Έναν constructor για τη φόρτωση της εικόνας και την προετοιμασία του αντικειμένου

Τις ίδιες μεθόδους απαιτούν και τα διαστημόπλοια των εξωγήινων! Η υλοποίηση θα διαφέρει σε ορισμένες μόνο μεθόδους. Επομένως, θα φτιάξουμε πρώτα μια γενική κλάση (υπερκλάση ή superclass) για όλα τα διαστημόπλοια και αργότερα θα δημιουργήσουμε εξειδικευμένες μεθόδους όπου χρειάζεται.

Υπερκλάση Craft

Σε μια πρώτη προσέγγιση, η κλάση για όλα τα διαστημόπλοια θα μπορούσε να γραφτεί ως εξής:

```
class Craft(object):
        def __init__ (self, imagefile, coord):
            self.shape = pygame.image.load(imagefile)
            self.ship_width = self.shape.get_width()
            self.ship_height = self.shape.get_height()
            self.rect = pygame.Rect(coord,(self.ship_width, self.
    ship_height))
        def Show(self, surface):
            surface.blit(self.shape,(self.rect[0],self.rect[1]))
        def Move(self,speed_x,speed_y, time):
            distance_x = speed_x * time
            distance_y = speed_y * time
            self.rect.move_ip(distance_x,distance_y)
        def Fire(self):
            pass
```

Αντί να χρησιμοποιήσουμε ξεχωριστές μεταβλητές για τις συντεταγμένες του διαστημόπλοιου, χρησιμοποιούμε ένα αντικείμενο της κλάσης Rect (από το rectangular, παραλληλόγραμμο). Αυτά τα αντικείμενα είναι λίστες με τέσσερα στοιχεία, της μορφής [Θέση_Χ, Θέση_Υ, Πλάτος, Ύψος]. Δείτε ένα παράδειγμα:

```
rect = pygame.Rect((240,560),(100,100))
```

Το καλό με την κλάση Rect είναι ότι περιέχει μεθόδους για να κινήσουμε το αντικείμενό μας. Ουσιαστικά, πρόκειται για μεθόδους οι οποίες μεταβάλλουν τις συντεταγμένες Χ και Υ. Επίσης, η ίδια κλάση περιλαμβάνει και μεθόδους για να ελέγχουμε αν ένα αντικείμενο βρίσκεται μέσα σε ένα άλλο (hint: αν το σκάφος έχει χτυπηθεί από βολή laser). Δείτε ένα απλό παράδειγμα για τη μέθοδο κίνησης:

```
self.rect.move_ip(distance_x,distance_y)
```

Ουσιαστικά, οι παραπάνω γραμμές αντικαθιστούν τις ακόλουθες δύο:

```
x = x + distance_x
y = y + distance_y
```

Αυτές τις δύο γραμμές τις έχουμε συναντήσει σε διάφορα προγράμματα, όπως το bouncing ball. Αν για κάποιο λόγο χρειαστούμε μόνο την τιμή του X ή την τιμή του Y, μπορούμε να την πάρουμε χρησιμοποιώντας τη σύνταξη rect[0] και rect[1], αντίστοιχα. Ένα σχετικό παράδειγμα φαίνεται και στο παραπάνω απόσπασμα κώδικα, στη μέθοδο Show:

```
        surface.blit(self.shape,(self.rect[0],self.rect[1]))
```

Η μέθοδος Fire, για την ώρα τουλάχιστον, περιέχει μόνο μια εντολή:

```
pass
```

Η εντολή δεν κάνει απολύτως τίποτα. Την έχουμε χρησιμοποιήσει μόνο και μόνο για να μην είναι κενός ο ορισμός της Fire. Προφανώς, δεν έχουμε σκεφτεί ακόμα το πώς θα λειτουργεί αυτή η μέθοδος! Η εντολή pass λειτουργεί σαν «place holder».

Από την υπερκλάση Craft θα δημιουργήσουμε τις κλάσεις για το δικό μας διαστημόπλοιο αλλά και για τους εξωγήινους. Για την ώρα θα φτιάξουμε μόνο την κλάση για το δικό μας σκάφος, αλλά μη νομίζετε ότι θα γράψουμε πολλά. Δείτε και μόνοι σας:

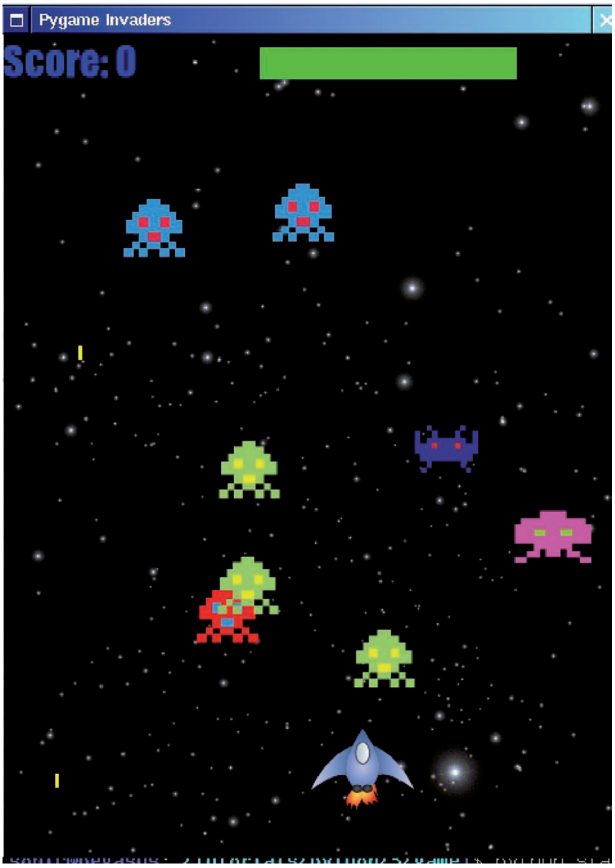
```
class SpaceCraft(Craft):
        pass
```

Όπως καταλαβαίνετε, έχουμε αφήσει πολλή δουλειά για αργότερα!

Κλάση για το φόντο

Έχουμε βρει μια ωραία φωτογραφία, με αληθινά αστέρια, την οποία θα χρησιμοποιήσουμε ως φόντο. Μελλοντικά θα την κάνουμε να σκρολάρει κατακόρυφα -- κι ελπίζουμε ότι μέσα στον πυρετό της μάχης, ο παίκτης δεν θα παρατηρήσει ότι επαναλαμβάνεται η ίδια εικόνα! Για την ώρα θα αρκεστούμε σε μια στατική απεικόνιση:

```
# Scrolling background
class SpaceBackground:
    def __init__(self,
imagefile):
        self.shape = pygame.
image.load(imagefile)
        self.coord = (0,0)
    def Show(self, surface):
        surface.blit(self.
shape,self.coord)
    def Scroll(self, speed_y,
time):
```



Εικόνα 3
Το παιχνίδι μας, όπως θα δείχνει στο (πολύ κοντινό) μέλλον. Παρατηρήστε τους retro εξωγήινους, οι οποίοι προέρχονται από το αρχικό Space Invaders, καθώς και το δικό μας (υπερσύγχρονο) διαστημόπλοιο. Τελικά, δεν είναι καθόλου τυχαίο το ότι δεν κυρίευσαν ποτέ τη γη. Eat my laser, alien!

```
        pass
```

Ο constructor δέχεται μόνο το όνομα αρχείου της εικόνας. Έχουμε φροντίσει το μέγεθος της εικόνας να είναι ίδιο με του παραθύρου. Οπότε, η απεικόνιση γίνεται στη θέση (0,0). Η μέθοδος Show, κατά τα γνωστά, εμφανίζει το φόντο επάνω στην επιθυμητή επιφάνεια. Η μέθοδος Scroll θα κάνει το φόντο να κυλάει κατακόρυφα και με την ταχύτητα speed_y. Αλλά αυτό αργότερα. Για την ώρα, το φόντο θα είναι στατικό.

Το κύριο πρόγραμμα

Έφτασε η ώρα να δούμε τον κώδικα για ολόκληρο το παιχνίδι. Βέβαια ο κώδικας είναι ακόμα αρκετά απλός. Αρχικά, ορίζουμε το μέγεθος του παραθύρου και στη συνέχεια τοποθετούμε το διαστημόπλοιό μας σε μία θέση:

```
pygame.init()
screenwidth,screenheight = (480,640)
spaceship_pos = (240, 540)
```

Στη συνέχεια, αφού ορίσουμε μερικά γενικά χαρακτηριστικά αλλά και τον τίτλο του παραθύρου, ρυθμίζουμε την ταχύτητα αντίδρασης του πληκτρολογίου:

```
screen = pygame.display.set_mode((screenwidth,screenheight),
DOUBLEBUF, 32)
pygame.display.set_caption(«Pygame Invaders»)
pygame.key.set_repeat(1,1)
```

Με το pygame.key.set_repeat(1,1), το πρόγραμμα θα αντιδρά το ταχύτερο δυνατό στα πατήματά μας! Στη συνέχεια χρησιμοποιούμε την κλάση για το φόντο και δημιουργούμε το αντικείμενο StarField. Δημιουργούμε επίσης το αντικείμενο SpaceShip, της κλάσης SpaceCraft. Πρόκειται για το δικό μας σκάφος και θα τοποθετηθεί στις συντεταγμένες που προσδιορίσαμε στη δεύτερη γραμμή (240, 540):

```
StarField = SpaceBackground(«stars.jpg»)
SpaceShip = SpaceCraft(«spaceship2.png»,spaceship_pos)
```

Αμέσως μετά δημιουργούμε το γνωστό αντικείμενο-ρολόι και εισερχόμαστε στον κύριο βρόχο του προγράμματος:

```
clock = pygame.time.Clock()
framerate = 60
while True:
    time = clock.tick(framerate)/1000.0
    shipspeed_x = 0
    shipspeed_y = 0
```

Στην αρχή του βρόχου μηδενίζουμε την ταχύτητα του σκάφους. Για να κινηθεί, πρέπει να πατήσουμε κάποιο πλήκτρο. Τότε το σκάφος θα μετακινηθεί κατάλληλα, θα ολοκληρωθεί ο βρόχος και η ταχύτητά του θα μηδενιστεί ξανά. Όταν δεν πατάμε κανένα πλήκτρο κίνησης τότε το σκάφος μένει ακίνητο, στην τελευταία γνωστή του θέση! Ο βρόχος συνεχίζει με τη διαχείριση των events:

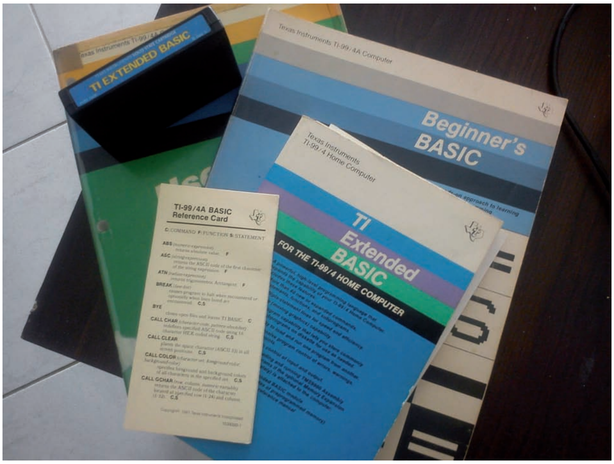
```
        for event in pygame.event.get():
            if event.type == QUIT:
                pygame.quit()
                exit()
            if event.type == KEYDOWN:
                key = pygame.key.get_pressed()
                if key[K_q]:
                    pygame.quit()
                    exit()
                if key[K_LEFT]:
                    shipspeed_x = -300
                if key[K_RIGHT]:
                    shipspeed_x = 300
```

Για τη διαχείριση των events, γενικά, χρησιμοποιούμε την pygame.event.get(). Ωστόσο όταν το event είναι τύπου KEYDOWN, καλούμε την pygame.key.get_pressed(). Αυτή η μέθοδος επιστρέφει μια λίστα με όλα τα πλήκτρα και μια τιμή True ή False για το καθένα, σύμφωνα με το αν πατήθηκε ή όχι. Έτσι, μπορούμε να διαβάσουμε περισσότερα από ένα πλήκτρα ταυτόχρονα! Αυτό είναι ιδιαίτερα χρήσιμο, διότι μπορεί να μετακινούμε το διαστημόπλοιο και την ίδια στιγμή να ρίχνουμε βολές. Το pygame.locals αντιστοιχεί συμβολικά ονόματα σε κάθε κωδικό πλήκτρου. Έτσι, μπορούμε εύκολα να ελέγξουμε το αν πατήθηκαν τα πλήκτρα που μας ενδιαφέρουν. Όπως αντιλαμβάνεστε, για την κίνηση του σκάφους έχουμε επιλέξει τα βελάκια. Αν πιεστεί το αριστερό βελάκι, η ταχύτητα του σκάφους αλλάζει σε -300, ενώ αν πιεστεί το δεξιό, η ταχύτητα γίνεται 300. Τέλος, ο βρόχος ολοκληρώνεται με τις ακόλουθες γραμμές:

```
        SpaceShip.Move(shipspeed_x, shipspeed_y, time)
        StarField.Show(screen)
        SpaceShip.Show(screen)
        pygame.display.update()
```

Εδώ, το διαστημόπλοιο κινείται με την ταχύτητα που εκφράζουν οι shipspeed_x και shipspeed_y. Η shipspeed_y είναι πάντα μηδέν, ενώ η τιμή της άλλης εξαρτάται από το αν πατήθηκε το αριστερό ή το δεξιό βελάκι. Κατά τα γνωστά, με τη βοήθεια της μεθόδου Show, εμφανίζουμε όλα τα αντικείμενα. Τέλος, για να προβληθούν στην οθόνη μας τα περιεχόμενα του frame buffer, καλούμε την pygame.display.update(). Απλούστατο, δε νομίζετε;

Είναι όντως απλούστατο, αλλά δεν είναι παιχνίδι! Που είναι οι εξωγήινοι; Που είναι οι βολές laser;



Εικόνα 4
Game programming στα 80s. Το Internet δεν υπήρχε και όλες οι πληροφορίες που μπορούσε να βρει κανείς για μια γλώσσα, περιέχονταν στο εγχειρίδιο του υπολογιστή! Εδώ βλέπετε τα βασικά βιβλία του TI-99/4A καθώς και το Extended BASIC cartridge, το οποίο κόστιζε μια μικρή περιουσία. Ευτυχώς, έβγαλε τα λεφτά του: Δουλεύει ακόμα!

Που είναι τα ηχητικά εφέ και η μουσική; Γι' αυτό το τελευταίο, θα κάνουμε κάτι αμέσως. Για τα υπόλοιπα θα χρειαστεί να περιμένετε μέχρι το επόμενο άρθρο. Όχι ότι θα σας αφήσουμε έτσι, βέβαια... Σας περιμένουν και οι ασκήσεις στο site του περιοδικού: http://deltahacker.gr/?p=5662

Μουσική

Για τη μουσική μπορούμε να γράψομε μια απλή συνάρτηση, την οποία θα καλούμε πριν από τον κύριο βρόχο του παιχνιδιού:

```
def PlayMusic(soundfile):
    pygame.mixer.music.load(soundfile)
    pygame.mixer.music.play(-1)
```

Η συνάρτηση δέχεται μία μόνο παράμετρο. Πρόκειται για το αρχείο μουσικής, το οποίο θα αντιστοιχηθεί στο (μοναδικό) κανάλι straming της pygame. Έτσι, για να αποκτήσει μουσική υπόκρουση το πρόγραμμά μας αρκεί να προσθέσουμε την παρακάτω γραμμή *πριν* από το ξεκίνημα του βρόχου:

```
PlayMusic(«spaceinvaders.ogg»)
```

Ευτυχώς, έχουμε προβλήματα!

Έχουμε γράψει μετά βίας 70 γραμμές κώδικα κι άρχισαν ήδη τα προβλήματα:

- Το σκάφος μας κινείται ανεξέλεγκτα. Κινείται μόνο αριστερά - δεξιά, αλλά μπορεί να βρεθεί εκτός οθόνης. Μπορείτε να κάνετε κάτι για να το περιορίσετε εντός;

- Μπορείτε να το κάνετε να κινείται και κατακόρυφα; Δηλαδή, να κινείται αριστερά, δεξιά, πάνω και κάτω. Φυσικά, πρέπει και πάλι να φροντίσετε ώστε να μένει πάντα μέσα στα όρια της οθόνης.

- Γράψτε μια συνάρτηση PrepareSound, η οποία θα δέχεται ένα όνομα αρχείου ήχου και θα επιστρέφει ένα αντικείμενο τύπου «sound». Σε αυτό το αντικείμενο θα έχει αντιστοιχηθεί το αρχείο που δηλώσατε και θα μπορείτε να το ακούσετε, χρησιμοποιώντας τη μέθοδο play του αντικειμένου. Όταν θα έχετε γράψει τη ζητούμενη συνάρτηση, θα μπορείτε να προσθέσετε στο πρόγραμμα γραμμές σας αυτές:

  ```
  explosion = PrepareSound(«explosion.wav»)
  explosion.play()
  ```

- Μπορείτε να εισάγετε έναν εξωγήινο (χρησιμοποιώντας την εικόνα alien1.png), ο οποίος θα εμφανίζεται στην πάνω αριστερή γωνία και θα σαρώνει την οθόνη; Ο εξωγήινος θα πρέπει να κινείται ομαλά προς δεξιά και όταν φτάνει στην άκρη της οθόνης θα πρέπει να επιστρέφει στα αριστερά, αλλά λίγο πιο χαμηλά. Η τροχιά του, δηλαδή, θα θυμίζει το γράμμα Z και θα τερματίζει στο ύψος όπου βρίσκεται το δικό μας σκάφος.

- Μπορείτε να σκεφτείτε κάποια μέθοδο για την κύλιση του φόντου;

Κάποια από τα παραπάνω ερωτήματα θα τα απαντήσουμε στο άρθρο 5.5, το οποίο θα βρείτε στο site του περιοδικού: http://deltahacker.gr/?p=5662.

Αυτός φυσικά δεν είναι λόγος για να μην προσπαθήσετε μόνοι σας! Έχετε τον κώδικα και γνωρίζετε πώς λειτουργεί. Το μόνο που σας λείπει είναι η καφεΐνη ;) Στο κάτω-κάτω, θα πρέπει να θυμόσαστε ότι το πραγματικό παιχνίδι για μας είναι ο κώδικας, σωστά;



Εικόνα 5
Το κάπως σκοτεινό desktop, ενός καθόλου σκοτεινού game programmer! Μπορείτε να καταλάβετε αν παίζει ή δουλεύει; Εκτός δηλαδή κι αν θεωρεί τη δουλειά παιχνίδι και το παιχνίδι δουλειά... Αλήθεια, σας γοητεύει μήπως το παράξενο desktop του υπολογιστή του και το θέλετε κι εσείς; Αν ναι μπορείτε να το έχετε -- και όλες οι πληροφορίες είναι στο http://deltahacker.gr/?p=5326

Skill: Intermediate

Tags: VPS, SSH, screen, rtorrent, BitTorrent, downloading



Έξυπνα, *γρήγορα* & άνετα κατεβάσματα, από το VPS!

by sub2raw

Με αρχή το τεύχος 005 του deltaHacker, εντελώς αθώα κι απρογραμμάτιστα γίναμε η αφορμή για να εκδηλωθεί ένα ζωηρό ενδιαφέρον γύρω από τους virtual private servers (VPS). Σας δείξαμε πώς να βρείτε και ν' αγοράσετε το δικό σας VPS, πώς να του εγκαταστήσετε λειτουργικό σύστημα και φυσικά πώς να το φέρετε στα μέτρα σας. Από εκεί και πέρα, οι εφαρμογές στις οποίες σταθήκαμε περιστρέφονταν κυρίως γύρω από το web hosting με τον lighttpd αλλά και την ασφάλεια, με το OpenVPN. Ε, σ' αυτό το τεύχος λέμε να συζητήσουμε για μια άλλη, εξαιρετικά ενδιαφέρουσα εφαρμογή που μπορεί να βρει το VPS σας ;)

Αν σας ρωτούσαν ποιο είναι το πλέον αγαπημένο χαρακτηριστικό ενός VPS, πιθανόν ν' αναφερόσασταν στο γεγονός ότι είναι εκεί, online και διαθέσιμο 24/7. Σωστό το σκεπτικό σας και, πράγματι, είναι σημαντικό να έχεις έναν ευέλικτο, σταθερό server εκτός του οικιακού δικτύου, ο οποίος είναι πάντα ανοικτός και προσβάσιμος απ' όλον τον πλανήτη. Το τι μπορεί να κάνει κάποιος μ' ένα τέτοιο box μόνο από τη φαντασία του περιορίζεται, όπως λέει και το κλισέ.

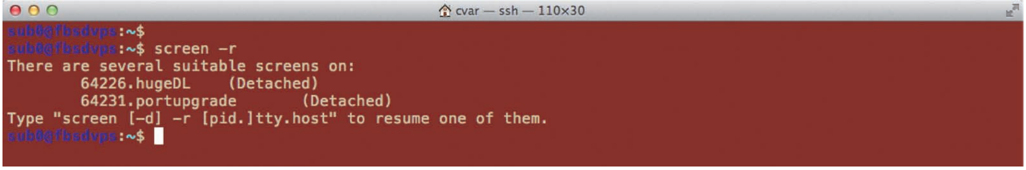
Ωραία. Κάποιο άλλο, αγαπημένο χαρακτηριστικό των VPS, μπορείτε να σκεφτείτε; Σίγουρα μπορείτε, αλλά για να μην το κουράσουμε πολύ ας το φέρουμε αμέσως στο προσκήνιο: είναι η ταχύτητα σύνδεσής του με το ίντερνετ. Όσο γρήγορη κι αν είναι η DSL σας, ακόμα κι ένα φθηνό VPS βγαίνει στον έξω κόσμο με πολλαπλάσια υψηλότερη ταχύτητα. Επιπρόσθετα, η ταχύτητα αυτή είναι συμμετρική: Όσο δηλαδή είναι το download speed, τόσο είναι και το upload speed. Ειδικά το τελευταίο, στις οικιακές συνδέσεις είναι πάντα ιδιαίτερα περιορισμένο (3Mbps ;h 5Mbps στην περίπτωσης μιας 36άρας ή 50άρας VDSL, 1Mbps στη συντριπτική πλειονότητα των περιπτώσεων). Μπορεί αρκετοί να μη δίνουν σημασία στο upload speed, κατά περιπτώσεις όμως η αξία του είναι τεράστια. Μας ενδιαφέρει, για παράδειγμα, τα torrents που κατεβάζουμε να τα μοιράζουμε με υψηλές ταχύτητες και με άλλους χρήστες, αν μη τι άλλο για να φτιάχνουμε καλό ratio όποτε χρησιμοποιούμε private tracker.

Καταλαβαίνετε, προφανώς, ότι το θέμα του διαθέσιμου bandwidth δεν το αναφέραμε στην τύχη. Είναι που σκεφτόμαστε ότι, πότε πότε, από άποψη ταχύτητας συμφέρει να κατεβάζουμε μέσα από το VPS και να αποθηκεύουμε στο δίσκο του. Για παράδειγμα, δύσκολα torrents που κατεβαίνουν αργά στον υπολογιστή μας και μοιράζονται ακόμα πιο αργά, σίγουρα είναι προτιμότερο να τα "παλεύουμε" από το VPS. Το ίδιο ισχύει και για μεγάλα downloads, τα οποία για τον άλφα ή βήτα λόγο θέλουμε να ολοκληρώνονται το συντομότερο δυνατόν. Άλλες φορές, πάλι, δεν θέλουμε να πνίγουμε τη γραμμή μας με torrents, ειδικά όταν χρησιμοποιούν κι άλλοι χρήστες του τοπικού δικτύου τη διαθέσιμη σύνδεση στο ίντερνετ.

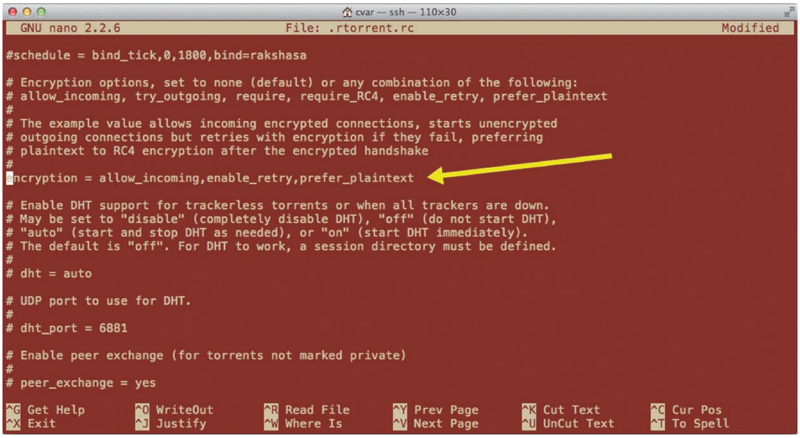
Θέλετε να σας αναφέρουμε κι άλλους λόγους για να *μην* κατεβάζετε torrents από το σπίτι; Πολύ ευχαρίστως! Κλείνετε όποτε θέλετε τον υπολογιστή και δεν φορτώνετε το λογαριασμό της ΔΕΗ, αποφεύγετε το bandwidth throttling που πιθανώς εφαρμόζει ο ISP σας, έχετε απόλυτο privacy για τα downloads σας, ενώ αποφεύγετε και τα όποια προειδοποιητικά mail από πλευράς ISP (κι όχι μόνο), περί απερίσκεπτης ή/και αμφιλεγόμενης χρήσης της σύνδεσής σας.

Κατεβάζοντας μέσα από το VPS, τίποτε από τα προηγούμενα δεν σας απασχολεί. Αφού τα downloads έχουν ολοκληρωθεί ή όταν, τέλος πάντων, τα θέλετε στον τοπικό υπολογιστή, ε, τότε θα τα μεταφέρετε από το VPS μέσως κάποιας ασφαλούς, κρυπτογραφημένης σύνδεσης (π.χ., μέσω SSH, SFTP ή ακόμη και με απλό FTP, αλλά μέσα από ένα OpenVPN tunnel μεταξύ τοπικού υπολογιστή και VPS[1]).

Έχοντας ονομάσει και τα δύο screen sessions, η λίστα που εμφανίζεται όταν δίνουμε screen -r προβάλλει τα PIDs αλλά και τα αντίστοιχα ονόματα. Έτσι, για τη σύνδεση σε κάποιο session μπορούμε να χρησιμοποιούμε το όνομά του :)



1. Περισσότερα για το OpenVPN και τα οφέλη του, καθώς και αναλυτικές οδηγίες εγκατάστασης και χρήσης, διαβάστε στο σχετικό άρθρο του deltaHacker 007.



Ενεργοποίηση της κρυπτογράφησης για το rtorrent. Εντάξει, ο ISP μας δεν έχει ιδέα για το τι κατεβάζουμε από το VPS, αυτό όμως δεν σημαίνει ότι ένα έξτρα επίπεδο προστασίας είναι ανεπιθύμητο.

Για να είμαστε απόλυτα ειλικρινείς και ξεκάθαροι, στην περίπτωση που αρχίσετε να χρησιμοποιείτε το VPS σας (και) για torrents υπάρχουν μερικά σημεία στα οποία οφείλετε να δώσετε ιδιαίτερη προσοχή -- αν μη τι άλλο επειδή κάποια από αυτά θα τα βρείτε μπροστά σας.

Είναι το bandwidth που σας παρέχει ο VPS provider απεριόριστο; Αν είναι (περίπτωση unmetered bandwidth), τότε έχει καλώς. Αν δεν είναι και το ξεπεράσετε τότε ο provider θα σας χρεώσει έξτρα -- και μάλλον θα σας χρεώσει πολύ. Το νου σας, λοιπόν: Αν έχετε ήδη VPS με metered bandwidth, κατεβάζετε με σύνεση. Αν πάλι ψάχνετε τώρα για VPS, προτιμήσετε ένα με unmetered bandwidth. Προσοχή: Διαφορετικοί providers εννοούν διαφορετικά πράγματα, όταν μιλούν για unmetered bandwidth. Για την ακρίβεια, ορισμένοι εννοούν "απεριόριστο bandwidth αλλά μόνον αν κατεβάζετε το πολύ τόσα GB ανά μήνα". Το ηθικό δίδαγμα είναι ότι πρέπει να προσέχετε τα λεγόμενα ψιλά γράμματα.

Πόσος είναι ο διαθέσιμος αποθηκευτικός χώρος του VPS σας; Ένας σοβαρός downloader θα θέλει τουλάχιστον 50GB, ωστόσο ο χώρος αυτός δεν βρίσκεται εύκολα σε φθηνά πακέτα VPS. Πάντως ακόμη κι αν οι απαιτήσεις σας σε αποθηκευτικό χώρο είναι υψηλές, αξίζει να το ψάξετε λίγο. Μάλλον θα εκπλαγείτε από τις προσφορές που θα βρείτε.

Πόσο καλά προστατεύεται η ιδιωτικότητά σας; Σίγουρα *πολύ* καλύτερα, σε σχέση με τα downloads από το σπίτι. Πράγματι, στην περίπτωση του VPS ο ISP σας δεν έχει ιδέα για το τι κατεβάζετε από εκεί, ενώ κι ο ίδιος ο VPS provider κατά πάσα πιθανότητα δεν θα νοιάζεται. Προσέξτε ωστόσο τους όρους χρήσης: Δεν είναι απίθανο το BitTorrent protocol να εμποδίζεται ή έστω να επιτρέπεται αλλά να απαιτείται σύννομη ή/και συνετή χρήση. Τώρα, εμείς δεν έχουμε κανέναν λόγο να πιστέψουμε ότι θα κάνετε παράνομη ή επιπόλαια χρήση. Οφείλουμε όμως να υπενθυμίσουμε ότι, στην περίπτωση που το παρακάνετε, ο VPS provider έχει τα στοιχεία σας και παρουσία εισαγγελικής εντολής δεν αποκλείεται να τα παραδώσει στις Αρχές. Αν λοιπόν δεν θέλετε να βρεθείτε προ απροόπτου, το λιγότερο που οφείλετε να κάνετε είναι να κατεβάζετε torrents δεχόμενοι μόνο κρυπτογραφημένες συνδέσεις, ενώ φυσικά ο συνδυασμός μέτρου και κοινής λογικής πρέπει να είναι σημαία σας.

Το υπόλοιπο του παρόντος άρθρου χωρίζεται σε τρία μέρη (όλα είναι στο ίδιο τεύχος, μην ανησυχείτε). Στα δύο πρώτα σας εξοικειώνουμε με

τα προγραμματάκια screen και rtorrent, τα οποία είναι απαραίτητα για downloading μέσα από ένα VPS. Το τρίτο μέρος είναι και το συντομότερο κι αφορά αφενός στο πώς στέλνουμε αρχεία στο VPS (π.χ., torrents που κατεβάζουμε πρώτα στον τοπικό υπολογιστή), αφετέρου στο πώς παίρνουμε αρχεία από το VPS (π.χ., ολοκληρωμένα downloads).

Πριν συνεχίσουμε, υποθέτουμε ότι τα προγράμματα screen και rtorrent είναι ήδη εγκατεστημένα στο VPS. Αν δεν είναι, εγκαταστήστε τα τώρα μέσα από τα αντίστοιχα εργαλεία διαχείρισης του λειτουργικού συστήματος. Για παράδειγμα, στο Ubuntu Linux αρκεί να πληκτρολογήσετε, διαδοχικά:

```
# sudo apt-get update
# sudo apt-get install screen rtorrent
```

Στο δε FreeBSD, μπείτε στον λογαριασμό του root κι αφού φρεσκάρετε το ports tree, π.χ., με

```
# portsnap fetch update
```

ακολούθως εγκαταστήστε το screen πληκτρολογώντας

```
# cd /usr/ports/sysutils/screen
# make install clean
```

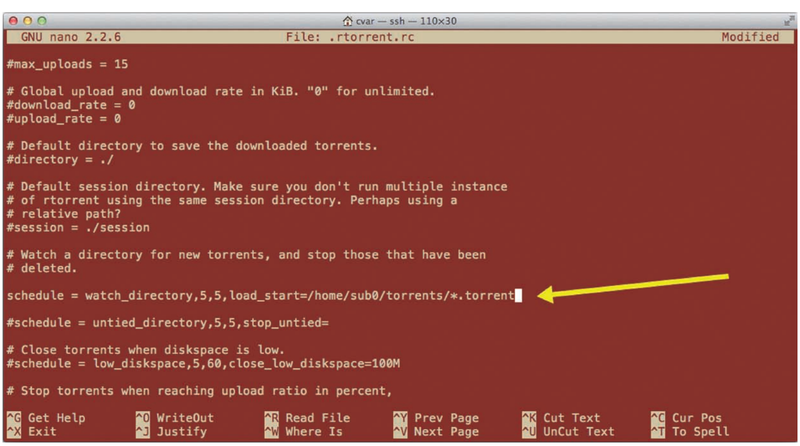
και μετά το rtorrent δίνοντας

```
# cd /usr/ports/net-p2p/rtorrent
# make install clean
```

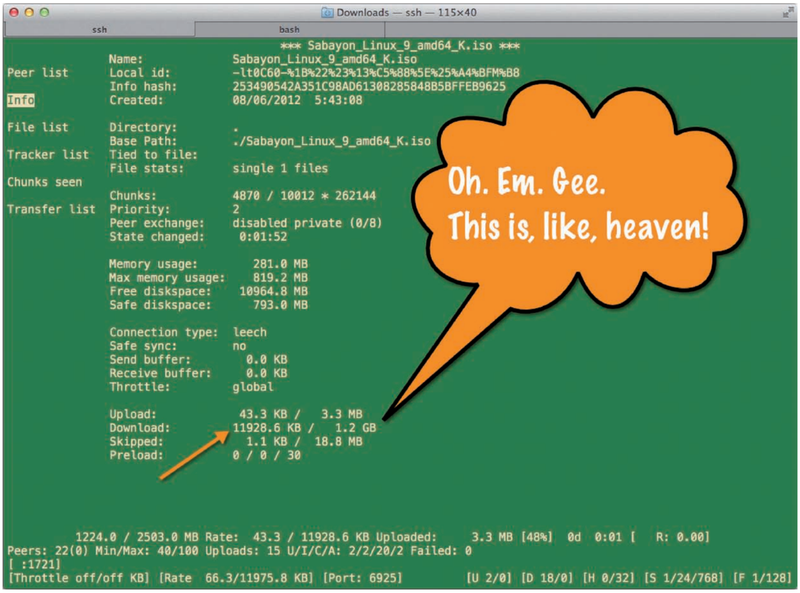
Είμαστε τώρα έτοιμοι για την πραγματικά ενδιαφέρουσα συνέχεια!

Μέρος 1 -- καλωσορίστε το screen!

Πολλές φορές η text based console του VPS αποδεικνύεται αρκετά περιοριστική για όσα θέλουμε να κάνουμε. Για παράδειγμα, αν θέλουμε να κατεβάσουμε κάτι με το wget, να τρέξουμε έναν torrent client, να κάνουμε μια μεταγλώττιση/εγκατάσταση και να παρακολουθούμε την κατάσταση του VPS με το htop, θα τα βρούμε σκούρα με μία μόνο κονσόλα. Έτσι καταφεύγουμε στην εύκολη λύση, δηλαδή αυτή της σύνδεσης και σε άλλες κονσόλες, φυσικά με ισάριθμα login, προκειμένου να απλώσουμε τις ερ-



*Μικρή επέμβαση στο αρχείο ρυθμίσεων ~/.rtorrent.rc, ώστε το rtorrent να κατεβάζει αυτόματα όσα torrents προσγειώνονται στον κατάλογο ~/torrents. Στο VPS μας, λοιπόν, μπορούμε να έχουμε πάντα ένα screen session ενεργό με το rtorrent να τρέχει μέσα του, και να είμαστε βέβαιοι ότι με το που θ' αντιγράφουμε ένα torrent file στον κατάλογο ~/torrents αμέσως θα ξεκινά το downloading. Προσοχή, όμως: Μιλάμε για VPS, οπότε δεν πρέπει να ξεχνάμε ότι ο διαθέσιμος αποθηκευτικός χώρος είναι περιορισμένος -- τουλάχιστον σε σύγκριση με ό,τι έχουμε συνηθίσει από τον υπολογιστή μας. Προσοχή χρειάζεται, επίσης, στις περιπτώσεις που ο VPS host *δεν* προσφέρει (αληθινό) unmetered bandwidth.*



Μπορείτε εσείς να πιάσετε αυτές τις ταχύτητες από το σπίτι;

γασίες. Εντάξει, είναι μια λύση κι αυτό, όμως τι γίνεται όταν θέλουμε να ξεκινήσουμε μερικές εργασίες (π.χ., downloads) και να απομακρυνθούμε από τον υπολογιστή μας, κάνοντας logout από το VPS ή/και κλείνοντας τον υπολογιστή που έχουμε μπροστά μας;

Για να μη χάνουμε το μπούσουλα με πολλές συνδέσεις, αλλά και για να μην εξαρτάμε τον υπολογιστή μας με τις όποιες ενεργές συνδέσεις προς το VPS, μπορούμε να καταφύγουμε σ' ένα μικρό και πανίσχυρο προγραμματάκι. Αναφερόμαστε στο screen, που διατίθεται για κάθε διανομή Linux/BSD.

Ας υποθέσουμε ότι έχουμε συνδεθεί στο VPS μας και θέλουμε να τρέξουμε ένα πρόγραμμα, που ξέρουμε ότι θα αργήσει να ολοκληρώσει την εργασία του. Το καλύτερο που έχουμε να κάνουμε είναι να τρέξουμε πρώτα το screen. Αμέσως η οθόνη θα καθαρίσει και θα μοιάζει σαν να πραγματοποιήσαμε μια νέα σύνδεση[2]. Από εκεί, αφού ξεκινήσουμε το επιθυμητό πρόγραμμα αρκεί να πατήσουμε [CTRL+A] και στη συνέχεια να πιέσουμε το πλήκτρο [D] (προσοχή: δεν εννοούμε τον συνδυασμό πλήκτρων [CTRL+A+D]). Με αυτόν τον τρόπο θα γίνει το λεγόμενο detach (εξ ου και το γράμμα «D»). Με απλά λόγια, το screen και το πρόγραμμα που τρέξαμε μέσα του θα αποδεσμευτούν από την κονσόλα μας και θα συνεχίσουν να εκτελούνται ανεξάρτητα από αυτή, στο background. Στη συνέχεια, αν θέλουμε μπορούμε να αποσυνδεθούμε από το μηχάνημα και να πάμε μια βόλτα. Η εφαρμογή μας --πάντα μέσα στο screen που την ξεκινήσαμε-- θα συνεχίσει την εργασία της κανονικά. Όταν επιστρέψουμε, είναι βέβαιο ότι θα θελήσουμε να δούμε την πρόοδο του προγράμματος που αφήσαμε να τρέχει. Για να γίνει αυτό θα μεταβούμε στο session του screen που αφήσαμε να τρέχει. Αφού συνδεθούμε λοιπόν στο VPS, θα πληκτρολογήσουμε απλά

```
# screen -r
```

2. Κάποιες φορές το screen είναι έτσι προρυθμισμένο ώστε να εμφανίζει μια οθόνη με ένα ενημερωτικό μήνυμα: για να φύγει πατάμε το [Space].

Έτσι, το screen session που είχαμε δημιουργήσει προηγουμένως θα έρθει και πάλι στο προσκήνιο κι εμείς θα μπορούμε να δούμε τι γίνεται... μέσα του! Πριν προχωρήσουμε παρακάτω, να σημειώσουμε ότι για τον τερματισμό ενός screen session αρκεί να συνδεθούμε σε αυτό (δίνοντας screen -r), να τερματίσουμε το όποιο πρόγραμμα τρέχει «εντός» και κατόπιν να πληκτρολογήσουμε exit ή να πιέσουμε το συνδυασμό πλήκτρων [CTRL+D].

Πολλά προγράμματα, σε πολλά screen

Όπως είδαμε, με τη βοήθεια του screen μπορούμε να αφήσουμε το VPS μας να εκτελεί μια εργασία *χωρίς* να είμαστε συνδεδεμένοι σ' αυτό. Όμως τι γίνεται αν επιθυμούμε την εκτέλεση πολλών προγραμμάτων; Πολύ απλά, δεν έχουμε παρά να ξεκινήσουμε πολλά screen sessions, τρέχοντας το screen όσες φορές χρειαζόμαστε. Πιο συγκεκριμένα, η διαδικασία έχει ως εξής: Ενεργοποιούμε το screen, ξεκινάμε την πρώτη εργασία και πατάμε [CTRL+A] και [D] (detach). Μετά τρέχουμε πάλι το screen, ξεκινάμε τη δεύτερη εργασία και κάνουμε πάλι detach. Συνεχίζουμε ομοίως για κάθε εργασία και στο τέλος μπορούμε να πάμε άλλη μια βόλτα, αφού βέβαια κάνουμε logout από το VPS. Οι εργασίες μας θα συνεχίσουν να εκτελούνται, η καθεμία στο δικό της screen session. Σε αυτή την περίπτωση, όταν αργότερα συνδεθούμε στο μηχάνημα και δώσουμε screen -r θα εμφανιστεί μια λίστα με τα screen sessions που εκτελούνται. Για να συνδεθούμε σε κάποιο εξ αυτών δεν έχουμε παρά να δώσουμε screen -r ακολουθούμενο από το PID (Process ID) του επιθυμητού screen session. Φυσικά, τα PID εμφανίζονται στη λίστα που αναφέραμε μόλις. Έτσι, μπορούμε διαδοχικά να συνδεθούμε σε κάθε screen, να ελέγξουμε την πορεία της αντίστοιχης εργασίας και, εφόσον το θέλουμε, να τερματίσουμε το αντίστοιχο session ή απλά να το αποσυνδέσουμε.

Ίσως κάποιοι σκεφτούν ότι η επανασύνδεση σε κάποιο screen, χρησιμοποιώντας το αντίστοιχο PID, δεν είναι ιδιαίτερα εξυπηρετική. Πράγματι, αν έχουμε ξεκινήσει πέντε-έξι screens, άντε να θυμηθούμε σε ποιο απ' όλα εκτελείται η τάδε εργασία που μας ενδιαφέρει. Η διαδοχική σύνδεση στο καθένα, εκτός από χρονοβόρα είναι κι εκνευριστική... Ευτυχώς, σε κάθε screen είναι δυνατόν να αποδίδουμε ένα όνομα. Έτσι, για τη μετέπειτα σύνδεση σε κάποιο session θα μπορούμε να γράφουμε το όνομά του (που θα εμφανίζεται στη λίστα με τα ενεργά screens) κι όχι το PID!

Η απόδοση ονόματος σε screen γίνεται με δύο τρόπους. Ο πρώτος είναι να δώσουμε το επιθυμητό όνομα κατά την εκτέλεση του screen. Για παράδειγμα, αν πληκτρολογήσουμε κάτι τέτοιο

```
# screen -S pcBSDdl
```

θα ξεκινήσει ένα νέο screen session με όνομα pcBSDdl. Αργότερα, όταν χρειαστεί να επανασυνδεθούμε στο συγκεκριμένο session, θα αρκεί να πληκτρολογήσουμε

```
# screen -r pcBSDdl
```

Αν ξεχάσουμε να βαφτίσουμε κάποιο screen την ώρα που το ξεκινάμε, δεν υπάρχει πρόβλημα. Ας υποθέσουμε ότι έχουμε ξεκινήσει ένα screen, έχουμε τρέξει κάποιο πρόγραμμα (μέσα στο screen) και ξαφνικά θυμηθήκαμε να ονομάσουμε το αντίστοιχο session. Το μόνο που έχουμε να κάνουμε είναι να πατήσουμε τον συνδυασμό [CTRL+A] και στη συνέχεια την άνω-κάτω τελεία [:]. Μ' αυτό τον τρόπο, στην τελευταία γραμμή του screen θα εμφανιστεί η άνω-κάτω τελεία και το πρόγραμμα θα περιμένει

να του δώσουμε κάποια εντολή. Ακριβώς τότε γράφουμε τη λέξη title και δίπλα το επιθυμητό όνομα. Στο τέλος πατάμε το [ENTER] και τελειώσαμε! Το screen έχει αποκτήσει το όνομα που γράψαμε και μπορούμε να κάνουμε detach.

Μέρος 2 -- καλωσορίστε και το rtorrent!

Το rtorrent είναι ένας λιτός στην όψη αλλά πλούσιος σε δυνατότητες BitTorrent client, ο οποίος λειτουργεί σε text based console. Διατίθεται για κάθε διανομή Linux, για όλες τις εκδοχές του BSD, καθώς και για το Mac OS X. Για τη συνέχεια υποθέτουμε ότι είναι ήδη εγκατεστημένο στο λειτουργικό του VPS σας.

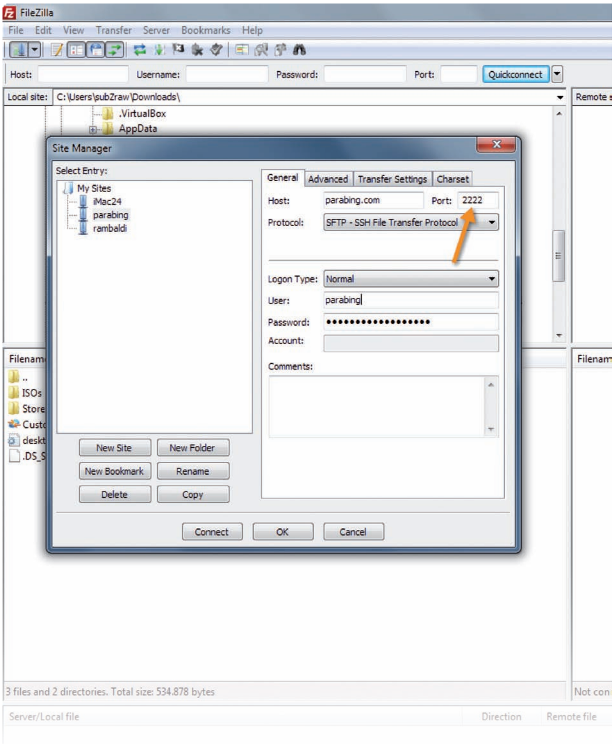
Την πρώτη φορά που θα τρέξετε το rtorrent, η λιτή του όψη είναι πιθανό να σας αποθαρρύνει. Όμως μη σταθείτε στη βιτρίνα: Στην πραγματικότητα, το rtorrent δεν έχει πολλά να ζηλέψει από προγράμματα σαν το μTorrent ή το Transmission. Πάντως πριν αρχίσετε να το χρησιμοποιείτε, αξίζει να εξοικειωθείτε με τα στοιχεία που εμφανίζει (αφού το τρέξετε, σε μια άλλη κονσόλα δώστε κι ένα man rtorrent). Υπάρχουν πολλοί τρόποι για να βάλετε το rtorrent να πιάσει δουλειά. Ο ένας είναι ο απευθείας, με μία μόνο εντολή. Ας υποθέσουμε ότι έχετε το torrent file με όνομα Sabayon_Linux_9_amd64_K.iso.torrent. Για να κατεβάσετε το αρχείο στο οποίο δείχνει, αρκεί να πληκτρολογήσετε στο τερματικό σας το εξής:

```
rtorrent Sabayon_Linux_9_amd64_K.iso.torrent
```

Εναλλακτικά, τρέξτε το rtorrent και πατήστε [Enter]. Στο κάτω μέρος του προγράμματος θα εμφανιστεί η γραμμή "load>". Εκεί, πληκτρολογήστε την πλήρη διαδρομή του torrent file ή εντοπίστε και συμπληρώστε το όνομά του με χρήση του [Tab], όπως ακριβώς θα κάνατε tab completion στο κέλυφος BASH. Πατήστε [Enter] και χρησιμοποιήστε τα πάνω/κάτω βελάκια, ώστε να φωτίσετε (highlight) το torrent. Ξεκινήστε το κατέβασμα πατώντας το συνδυασμό πλήκτρων [CTRL+S]. Ακολουθήστε παρόμοια διαδικασία για να προσθέσετε κι άλλα torrent files στην ουρά. Προκειμένου να σταματήσετε ένα torrent, φωτίστε το με τα βελάκια και δώστε [CTRL+D]. Αν το φωτισμένο torrent είναι ήδη σταματημένο και πατήσετε πάνω του [CTRL+D], τότε θα διαγραφεί. Εξάλλου, αν κάποια στιγμή θελήσετε να τερματίσετε το rtorrent, κάντε το μ' ένα [CTRL+Q]. Για περισσότερες συντομεύσεις πληκτρολογίου συμβουλευτείτε τον σχετικό πίνακα.

Σε κάθε περίπτωση, τα δεδομένα που κατεβάζει το rtorrent τα αποθηκεύει στον κατάλογο όπου βρι-

Το FileZilla είναι ένας άριστος, δωρεάν FTP client που διατίθεται για όλες τις δημοφιλείς πλατφόρμες. Μεταξύ των πρωτοκόλλων που υποστηρίζει είναι το SFTP (FTP μέσω SSH) και είναι ό,τι πρέπει για να μεταφέρουμε τα downloads από το VPS στον τοπικό υπολογιστή. Στη φωτογραφία φτιάχνουμε ένα νέο entry στον Site Manager του FileZille, για ένα VPS που έχουμε στο New Jersey. Παρατηρήστε ότι για λόγους ασφαλείας ο SSH server του VPS δεν τρέχει στο προκαθορισμένο port, που είναι το 22.



σκόσασταν την ώρα που το τρέξατε. Αυτή η συμπεριφορά εύκολα μπορεί ν' αλλάξει και να γίνει πολύ πιο ενδιαφέρουσα. Κατ' αρχάς, μπορείτε να ορίσετε έναν κατάλογο όπου θα ελέγχει το rtorrent την πρώτη φορά που το τρέχετε ή εφόσον τρέχει. Με το που θα βάζετε εκεί ένα torrent file, το πρόγραμμα θα αρχίζει αυτόματα το κατέβασμα. Αν πάλι διαγράψετε κάποιο torrent file από κει, το rtorrent θα σταματάει το κατέβασμα των αντίστοιχων δεδομένων, επίσης αυτόματα. Μπορείτε, επίσης, να υποδείξετε τον ίδιο ή έναν άλλον κατάλογο, όπου εκεί θα αποθηκεύονται τα δεδομένα στα οποία δείχνουν τα torrent files. Αυτές κι άλλες παραμέτρους της συμπεριφοράς του rtorrent μπορείτε να τις καθορίσετε από το αρχείο ρυθμίσεων του προγράμματος, το ~/.rtorrent.rc. Μη διστάσετε να πειραματιστείτε μ' αυτό το αρχείο. Ακόμα και τα φώτα να του αλλάξετε, πάντα θα υπάρχει ένα πρότυπο στο οποίο θα μπορείτε να επανέλθετε. Το πού είναι και πώς ονομάζεται αυτό το πρότυπο έχει να κάνει με το λειτουργικό. Στο Ubuntu, π.χ., βρίσκεται στον κατάλογο /usr/share/doc/rtorrent/examples, ενώ στο FreeBSD είναι στον κατάλογο /usr/local/share/examples/rtorrent. Περισσότερα για τις δυνατότητες του rtorrent μπορείτε να διαβάσετε πληκτρολώντας man rtorrent. Την ώρα που διαβάζετε το manual page, πάντως, μην αφήνετε το πρόγραμμα να τεμπελιάζει ;)

Συντομεύσεις πληκτρολογίου για το rtorrent

[CTRL+Q]	Τερματίζει το πρόγραμμα
[CTRL+S]	Αρχίζει το κατέβασμα του φωτισμένου (highlighted) torrent
[CTRL+D]	Σταματά το ενεργό torrent ή το διαγράφει, αν είναι ήδη σταματημένο
[Πάνω], [Κάτω]	Πλοήγηση στα torrents της ουράς, φωτισμός του τρέχοντος
[Enter]	Βάζει ένα νέο torrent στην ουρά, χωρίς να ξεκινήσει το κατέβασμά του
[Backspace]	Βάζει ένα νέο torrent στην ουρά και ξεκινά αμέσως το κατέβασμα
[+], [-]	Αλλαγή της προτεραιότητας του φωτισμένου torrent
[a] / [s] / [d]	Αύξηση της καθολικής ταχύτητας upload κατά 1 / 5 / 50KB/s
[z] / [x] / [c]	Ελάττωση της καθολικής ταχύτητας upload κατά 1 / 5 / 50KB/s
[A] / [S] / [D]	Αύξηση της καθολικής ταχύτητας download κατά 1 / 5 / 50KB/s
[Z] / [X] / [C]	Ελάττωση της καθολικής ταχύτητας download κατά 1 / 5 / 50KB/s

Ένα ερώτημα που πιθανώς σας δημιουργήθηκε έχει να κάνει με τη μεταφορά εκείνου του torrent file του παραδείγματος, στο VPS. Πώς το πήγαμε εκεί; Επίσης, από τη στιγμή που κατεβάσουμε κάτι στο VPS, πώς το παίρνουμε στον τοπικό υπολογιστή; Οι απαντήσεις είναι στο τρίτο και τελευταίο μέρος του παρόντος. Συνεχίστε το διάβασμα.

Μέρος 3 -- μεταφορές από και προς το VPS

Όσοι αισθάνονται άνετα με τη γραμμή εντολών και στον υπολογιστή τους τρέχουν Linux, BSD ή OS X, τις πρώτες μεταφορές αρχείων από και προς το VPS θα τις κάνουν μέσω του ασφαλούς πρωτοκόλλου SSH και συγκεκριμένα με τη βοήθεια του εργαλείου scp. Ιδού ένα παράδειγμα:

```
$ scp -P 2222 ~/Downloads/Sabayon_Linux_9_amd64_K.iso.torrent \
sub0@fbsdvps.thruhere.net:~/
```

Παρατηρήστε την παράμετρο -P, η οποία είναι προαιρετική και υποδηλώνει το port που χρησιμοποιεί ο απομακρυσμένος SSH server (στο VPS). Εξ ορισμού, το port για το SSH είναι το 22/TCP. Στο παράδειγμα, ωστόσο, για λόγους ασφαλείας είναι αλλαγμένο σε 2222. Ο λογαριασμός χρήστη στο VPS, στον οποίο αντιγράφουμε το αρχείο Sabayon_Linux_9_amd64_K.iso.torrent, είναι εκείνος του sub0. Το πλήρες domain name που αντιστοιχεί στο VPS είναι το fbsdvps.thruhere.net και του το χαρίσαμε με τη δωρεάν υπηρεσία dynamic DNS του http://dyn.com (εναλλακτικά, δείτε και το www.no-ip.com). Αν δεν το είχαμε κάνει, τότε στη θέση του fbsdvps.thruhere.net θα βάζαμε το (στατικό) IP του VPS. Δείτε κι ένα παράδειγμα αντιγραφής ενός ολόκληρου καταλόγου (φακέλου), από το VPS στο τοπικό μηχάνημα:

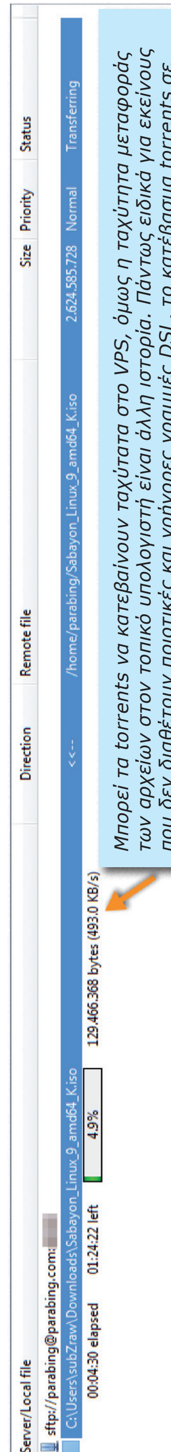
```
$ scp -P 2222 -r sub0@fbsdvps.thruhere.net:~/ISOs ~/Downloads/
```

Το καινούργιο εδώ είναι η παράμετρος -r (από το recursive), η οποία προκαλεί τη λήψη του φακέλου ISOs και όλων των περιεχομένων του. Περισσότερα για το scp θα μάθετε από το αντίστοιχο man page (δώστε man scp).

Πέρα από τη γραμμή εντολών, έχετε τη δυνατότητα να χρησιμοποιείτε έναν οποιονδήποτε FTP client για το περιβάλλον γραφικών (καλό είναι να υποστηρίζει και συνδέσεις SFTP, δηλαδή FTP πάνω από SSH). Το FileZilla Client είναι μια άριστη επιλογή για Windows, OS X και Linux. Μπορείτε να το κατεβάσετε από το http://filezilla-project.org. Μια άλλη, δωρεάν λύση για Windows και OS X, είναι το Cyberduck (http://cyberduck.ch).

Στην περίπτωση, τέλος, που στο VPS σας τρέχει ο OpenVPN server και είστε συνδεδεμένοι σ' αυτόν (βλ. σχετικό άρθρο στο deltaHacker), τότε μπορείτε να μεταφέρετε αρχεία από και προς το VPS με απλό FTP (χωρίς κρυπτογράφηση) ή ακόμη και με drag'n'drop μεταξύ τοπικού μηχανήματος και VPS -- αρκεί βέβαια στο τελευταίο να έχετε εγκατεστημένη και ρυθμισμένη τη SAMBA. Αν πάντως μιλάμε για μεγάλα αρχεία, τότε αντί για τη SAMBA είναι σαφέστατα προτιμότερο να χρησιμοποιείτε (S)FTP: Σε περίπτωση που για κάποιο λόγο η σύνδεση διακοπεί, με τη SAMBA *δεν* θα μπορέσετε να συνεχίσετε τη μεταφορά από εκεί που είχε μείνει και θα πρέπει να ξεκινήσετε από την αρχή. Να σημειώσουμε ότι στην περίπτωση του VPN, προσέξτε ώστε οι μεταφορές να γίνονται μέσω του κρυπτογραφημένου τούνελ. Πρακτικά, αυτό σημαίνει ότι θα συνδέεστε στο IP που είναι στο απομακρυσμένο άκρο του tun device (συνήθως το 10.8.0.1 -- αφού έχει επιτευχθεί η σύνδεση στον OpenVPN server).

Παίξτε με τα προγράμματα που σας προτείναμε, δηλαδή με το screen και το rtorrent, δαμάστε τα κι αρχίστε να πειραματίζεστε με downloads στο VPS. Πιστεύουμε --για την ακρίβεια ξέρουμε-- ότι θα υπάρξουν περιπτώσεις κατά τις οποίες θα χαίρεστε ιδιαίτερα που αποφασίσατε να χρησιμοποιείτε το VPS σας *και* για τέτοιες δουλειές.



Μπορεί τα torrents να κατεβαίνουν ταχύτατα στο VPS, όμως η ταχύτητα μεταφοράς των αρχείων στον τοπικό υπολογιστή είναι άλλη ιστορία. Πάντως ειδικά για εκείνους που δεν διαθέτουν ποιοτικές και γρήγορες γραμμές DSL, το κατέβασμα torrents σε VPS αποτελεί αξιόπιστη λύση η οποία, την ίδια στιγμή, προσφέρει ασφάλεια, προστατεύει την ιδιωτικότητα και αφήνει τη γραμμή ελεύθερη για άλλες εφαρμογές/χρήσεις.

Εσείς μάθατε για την **προσφορά** του Ιουνίου 2012;!

Αγοράστε τώρα μια 12μηνη ή 24μηνη συνδρομή στο deltaHacker ή προσθέστε στη συνδρομή που έχετε άλλη μια 12μηνη ή 24μηνη συνδρομή, και αποκτήστε αυτόματα **5% έκπτωση** επί της κανονικής τιμής!

Επιπλέον, όσοι επιλέξετε 24μηνη συνδρομή μπαίνετε και σε κλήρωση που θα γίνει στις 6/7/2012, για πέντε (5)

συνδρομές εφ' όρου ζωής!

Ναι, σωστά διαβάσατε: Οι πέντε τυχεροί που θα κληρωθούν θα παίρνουν το περιοδικό για πάντα!

Για να εκμεταλλευτείτε την προσφορά του Ιουνίου αγοράστε την 12μηνη ή 24μηνη συνδρομή σας **έως και τις 30/6/2012**, συμπληρώνοντας τη φόρμα παραγγελιών στο

http://deltahacker.gr/order

και στο πεδίο με τα σχόλια γράψτε

έκπτωση 5%

Προσοχή: ακολουθούν ψιλά γράμματα!

Η έκπτωση του 5% και η κλήρωση για τις πέντε συνδρομές εφ' όρου ζωής, ισχύουν μόνον για όσους αγοράσουν 12μηνη ή 24μηνη συνδρομή **έως και τις 30/6/2012**.